Universitat Politècnica de Catalunya
Departament de Llenguatges i Sistemes Informàtics
Màster en Intel.ligència Artificial

Master Thesis

Using AI techniques to determine promoter location based on DNA structure calculations

Carlos Fenollosa Bielsa

Advisors:

Ramon Goñi Macià, MMB, PCB
Javier Vázquez-Salceda, LSI, UPC

Barcelona, September 2008

"To build computers that are like humans, we first need to understand how we were designed. Bad thing, as we were never designed, but evolved from the chaos with no documentation left."

Abstract

DNA sequencing projects have started the race to fully annotate complete genomes, including the human one. Despite that, little is known about genetic regulation, the mechanisms that control where and when the genes are expressed, and promoters are maybe the most important of these mechanisms.

An increasing number of studies have been focused on the DNA molecule and its structure. This has lead to a set of physical properties which can be computed from mathematical models, and describe some aspects of this molecule. Unfortunately, the existing tools are scattered through the different web sites of many research groups, and extracting data with them is still very unpleasant. The first part of this thesis presents DNAlive, a new platform to calculate DNA physical properties, showing the results in a visual and useful way for genetic researchers, cross-linking the data with external databases.

For the second part, a full study of DNA physical descriptors has been performed, revealing significative similarities between them. Using that data, a set of neural networks has been developed to detect promoters on a DNA sequence. The resulting software is the second version of ProStar, the MMB group's[1] latest promoter predictor.

[1]The *Molecular Modelling & Bioinformatics Group* (`http://mmb.pcb.ub.es`) is a research group hosted at the Parc Científic de Barcelona. I have been working there since 2007, in a joint programme with the Barcelona Supercomputing Center, and this MSc Thesis has been developed as one of the group's projects.

Acknowledgements

This Thesis could be summed up as *Structural biology and genetics for computer scientists: From nothing to "a lot" in just a year*. That is why these acknowledgements are the most important part of *this* thesis.

First of all, I need to thank Ramon Goñi for offering me a career in the Life Sciences, for his invaluable help and guidance since day one, for his interest and dedication, for calling me from Madrid whenever it was needed and sending me e-mails at late night hours. You reviewed my work as if it were your own responsibility. Thanks for that.

On the other side, I'm sure that Javier Vázquez has also learned a lot of biology by brute force. It's not easy to understand genes and promoters, let alone if the teacher is me. Everything would have been easier if I had chosen the blue pill named Contract, but I love challenges. Thanks, and sorry!

I'd also like to thank Modesto Orozco for getting time from nowhere to meet with me and teach me concepts from Biology 101. Maybe he would get the Nobel Prize if he finally admitted that he has discovered 30-hour-long days.

At the LSI, Karina Gibert and Lluís Belanche offered me great help on the PCA and the neural network. I still keep the napkin from the cafeteria where Karina drafted the matrix I needed to build to run the PCA.

Thanks to José Antonio Alcántara for his fast help with all my server issues. A special acknowledgement also for the people at the INB, and I mean all of them, for explaining me the internals of BioMoby, which are not always nice. Many eyes have been upon DNAlive and have helped me to debug the whole platform, so thanks to the people in the MMB group in general and the web testers in particular, especially David Torrents at the BSC.

To all my workmates, I'm glad I've met you! It is a pleasure to work with you and a privilege to have those profound conversations from time to time. In no particular order, but ladies first: Montse, Laura, Jordi and Felix, with occasional outsiders.

Finally, as I don't want to leave anybody out of these acknowledgements, here's a placeholder: thanks to `$you`[2] too!

But wait, there's more!

To my mom: it's okay to be stressed when there's only a week left to hand the manuscript in. Don't try to calm me down, I need the stress to stay awake. You are guilty for teaching me curiosity and ambition, paying my University so that I could study on weekends, I don't want to screw it up in the last minute.

[2] PHP joke. Not Perl. PHP.

I'm fine now. I'm sure that grandpas will be very happy because I have finished, at last.

To my friends: It's okay not to go out saturday nights if you need to stay until five o'clock AM running batches in the computer nodes. I don't want to disconnect and relax for a couple of hours because I'll feel guilty afterwards, but I want to tell you that I appreciate the effort. Really. Remember how you were so worried that you didn't allow me to attend your PFC defense? We can party afterwards (and we will).

This Thesis also accounts as my PFC for Computer Engineering. Seven long years, the latter two attending the MSc, have allowed me to meet a lot of cool people at the FIB. Some carried on, some dropped out, some switched to the technical engineering and finished in only three years. As I love writing —I think you have already noticed it by the length of this section— on day zero I joined l'Oasi, a student group at the FIB that publishes a magazine every semester. This was the best decision I made during my studies, because I found the boldest people in college there. l'Oasi has been my second home for five years.

But wait, there's even more!

The acknowledgements finish here, however. If you want to keep reading, you will find a personal analysis on everything regarding my experience on this thesis. Not the scientific conclusions, not textbook-like information, but my own experience. I suppose that I could have posted it to my blog instead, but I want it attached to this document. You have been warned.

When you do blind data mining, you learn something: data is just data. The trick is how to interpret it. But how is a computer engineer supposed to discover whether the correlation between something named "a-philicity" and "propeller twist" *looks good* or not?

Only after a year of working in bioinformatics I am able to slightly understand the data I'm working with. For so many time it was just numbers that only took shape when shown to my advisors. I was working blindly. I could only discover that the algorithm for calculating correlations had a bug when Ramón told me that the results "didn't look good".

Working in science has something special. You could get disastrous results only to discover later that the methods are OK but there is a bug in a small module of your software. And sometimes, even published papers contain erroneous data that will be refuted years later. Science is about discovering things, things that nobody else but you is working on. Results that nobody can tell whether they are correct or not, because everything is so new. Science is not math, biology is not a computer algorithm which will output a one or a zero.

I thought that neural networks are nice because they almost do magic. They can classify anything, and sometimes pretty good. Unfortunately, that turns out to be true only for the Iris.arff[3] dataset and a couple more which a toddler could linearly separate with a crayon and a ruler. In biology, data has a lot of noise, the datasets might be not 100% trustworthy and we don't even know how to start modeling what happens in our body. Did I mention that we can extract huge amounts of information that can't be treated computationally?

[3]Machine-learning joke

As an example, the gene expression is so tricky that there could be a gene hidden in some obscure region of the DNA and we don't know of its existence because it expresses only in the liver, and only in the first month of pregnancy. Then, it becomes silent forever. How are we supposed to discover it? Do we need to make a map of all the expressed DNA in all organs for every 10 minutes since the first second of gestation? The answer is even worse than the question: yes, we need; and no, we don't know how to generate this data. Or where to store and process it.

But it turns out that a computer guy can learn biology. Fortunately, all the institutions that form the MMB group have weekly seminars —this sometimes means more than five seminars a week— where everybody can learn from every field. Some are boring, some are interesting, some are so specific that you only understand them some months later, after working a bit on that field and reading a lot of Wikipedia articles[4]. But, in the end, you learn something.

It's funny because biologists and chemists also tell you their own challenge: learn to use a computer and program software. They might have a PhD, but some only know enough computer science to run MS Word in Windows, because they've always worked in a lab. The solution? Collaborate. They explain me what a propeller is, and I explain them how to Perl[5]. Sadly, from time to time I am asked a question on FORTRAN, and I can't help. In the end, it is all about mixing experiences, and it is very rewarding.

I studied computer engineering because I wanted to know more about computers. And, thankfully, when the major is finished, computers have no secrets. One can explain the whole process between pressing a key on a laptop and getting a map from Google, step by step. Unfortunately, we can explain less things about men, society and the universe. We need knowledge, collaboration, computers, and further developments in Artificial Intelligence. Nowadays, a giant leap in computer architectures is required, because even with huge artifacts like the Mare Nostrum, Turing Machines are basic sequential machines which are not that good for processing real world data.

That being said, if you have reached this end line, the final acknowledgement goes for you. Thanks for reading.

[4]How could we live without Wikipedia?

[5]It seems that a lot of bioinformatic libraries are written in Perl. I also invest some time on trying to convince them to switch to PHP for their own scripts, with mixed results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Sequencing projects have provided the genome sequence of many evolved organisms, including mammals. Unfortunately, less information exists on the detailed mechanisms controlling gene expression.

Promoter[1] and gene detection is one of today's biggest bioinformatic problems because of its complexity. Genes control all the functions of the human body, but the models found to date do not work properly for some kind of genes. Furthermore, it is a recent field of study, and very important research groups are trying to tackle this problem with bioinformatic tools. The use of AI techniques over statistical models can help researchers to find genetic models, even if they cannot be fully understood.

Despite the impressive success of high-throughput experimental techniques, the determination of promoters is still a big challenge in the post-genomic era. Promoter prediction would be straightforward if all genes were perfectly annotated and promoters were always placed at the same relative location respective to the gene.

Unfortunately, annotation is often subject of large uncertainties and recent genomic analysis has shown that the naive concept of a gene having a single start point located near the gene is not correct. Furthermore, one gene region might have several Transcription Start Sites, one promoter might induce transcription start at different sites and promoter regions often overlap [Cea05]. On the other hand, sequence signals associated to promoters are rather unspecific, generating a large percentage of false positives [BSC+02]. In other words, theoretical prediction of promoters is still one of the greatest challenges in bioinformatics.

Bioinformatics is a recent field of study, and involves the use of computers and algorithmic knowledge to solve biological problems. In the last 30 years, thanks to the huge improvements in computers and databases, biologists have started to use them to cross biologic data with the hope of finding new information. Later on, with the introduction of multidisciplinary researchers, the data was converted to information structures and now almost every biological field can use computers to enhance their work.

On the interface part, software tools for bioinformatics range from simple command-line tools, to more complex graphical programs and standalone web-services. With the increasing availability of Internet tools and the invention of

[1] The reader might find in this introduction some biological terms. All of them are explained in chapter 2.

remote data exchanging protocols, it is now possible to run software in remote servers, obtain the results, parse them, and send the results to another machine which will display them in a graphical window.

Databases are huge and scattered, and there is room for improvement in data display interfaces as well as new methodologies to extract more information from that data. The trend is moving from using only one algorithm and one interface —a terminal running a script— to the use multiple sources of information and multiple interfaces.

1.1 Definition of the problem and objectives

Despite the recent sequencing of the **human genome**[2] [VAM+01], biologists know little about the mechanisms regulating the genes. One of the most important regulators are promoters, regions of DNA located near the genes. Specific proteins recognize these promoter areas and start the transcription of the gene.

Promoters can be discovered by experimental means, in vitro, but these methods are slow and difficult. Since the sequencing of the human genome, early bioinformatic tools have been trying to discover new promoters [DGZ01], changing the computational approach with every new genetic discovery.

The first algorithms [OcLNR02] relied on the DNA sequence itself, while the most recent techniques [ASRdP08] use structural profiles of the DNA as the base for their predictions. In this Thesis, we will follow the most recent trends on promoter detection, designing a platform to calculate structural profiles from a DNA sequence, and then a promoter detection software which will use these profiles to predict promoter location.

In the first part, we will implement DNAlive, a multi-interface platform to compute up to 29 physical descriptors of a DNA molecule (listed in appendix. B). For the second part, following the community-standard EGASP [BBB+06] experiment, we will analyze the DNA profiles of a public DNA database and use Neural Networks to improve the prediction power of a state-of-the-art software developed by the MMB group, ProStar.

1.2 Application of the AI

The first part of the Thesis, the construction of DNAlive, is mainly a computer engineering problem. It consists on the development of two interfaces to allow the remote execution of DNA descriptors. The interfaces are a web page and bioinformatic web services.

To implement the descriptors, two basic algorithms were used: geometric operations on the input data and fuzzy logic, where the resulting scalar value of an operation could be modified by conditional distributions. However, the methodology for each one is clearly specified on their respective paper (see appendix B). Thus, even in the cases where some AI technique is used, it was not developed during this thesis, and as such it is not explained here.

On the second part, ProStar 2 makes extensive use of AI methodologies. The core techniques that will be used to improve the previous version, and detailed in ch. 5, are:

[2] All terms in **bold** are included in an Index at the end of the document.

- Analyze the DNA descriptors and determine which are more valuable to detect promoters, using data mining techniques.
- Promoters will be classified in four groups, each one representing a different promoter type. Then, each one will be treated separately to gain specificity.
- A neural network for each group will be trained and then used to predict the presence of a promoter in a new DNA sequence.

The promoter detection task has usually been labeled as a symbolic problem. The DNA sequence in its naked form was the only information used, and each element of the sequence is a symbol (A, C, G, T). Classic techniques include pattern matching or stemming, or even Hopfield networks to reduce noise.

By using DNA numerical descriptors instead of the plain sequence, the problem turns to be subsymbolic. After computing the descriptor, the original sequence is discarded, only the numerical values are kept. The use of this method allows to plot a graph of the descriptor signal along the sequence, and some important DNA elements can be discovered by looking at the signal. Unfortunately, this is not yet the case for promoters.

The descriptors analysis will help to reduce the search space and input noise. These values have biological meaning, but it is unknown which one of them best describes a promoter. Using a comparison, looking for promoters by using the DNA temperature value could be like trying to define a musical composition by the color of the speaker it plays on.

Statistical methods will be used to compute the relevance of each descriptor, starting with a Pearson correlation and then running a Principal Components Analysis. Correlations will tell which descriptors are redundant, and the PCA will determine the weight (i.e. relevance) of each descriptor.

Promoters can be clustered in different ways, based on some of their properties. In this thesis, they will be classified by analyzing the presence or absence of the two most common elements, forming four groups. The classification is quite simple, and one interesting conclusion will be whether these groups are differentiated by physical descriptors or not.

In order to get a class predictor (promoter or non-promoter), many classifiers could be used. For this problem, neural networks were chosen, as it is a very flexible method, specially suited to treat sets of inputs as signals and to build models that are, to some extent, noise-resistant.

Other alternatives, such as support vector machines, were discarded because the lack of a kernel function. Self organizing maps are used in [ASRdP08] but, as we are going to cluster the promoter instances beforehand, it makes no difference compared to a neural network.

Other AI techniques were discarded because the lack of application. Some types of learning, like reinforcement learning, cannot be applied, as the answer space is not big enough. In a same manner, it is impossible to use deductive learning or rule-based learning, because there is little knowledge about the solution, no model has been found to define a promoter, and thus the model has to be created.

This thesis is a clear example of the application of the AI in many research fields. In bioinformatics, promoter prediction is a main problem which will lead to better genetic research.

The only way to validate the predictions is to compare them to an annotated DNA database, but these databases are very small. When a predictor performs well, research groups use it to find places where it detects unknown promoters with the most confidence value, and then look for them in vitro, in a biology laboratory.

Not only is it important to discover promoters, but also improve the confidence values and reduce the number of false positives. This thesis' results will provide light on the application of new methodologies to a current problem.

1.3 Structure of this thesis

This document is organized as follows. After the introduction, in chapter 2 we introduce the biological concepts behind this thesis. In chapter 3, current bioinformatics software will be presented, plus a series of gene predictors, including the first version of ProStar.

Chapters 4 and 5 explain the methodology that we have used to implement the DNAlive platform and ProStar 2 promoter predictor, respectively. After presenting the methods, chapter 6 shows the results for both projects, and chapter 7 analyzes these results, comparing them to the initial objectives.

Two appendices are included. Appendix A contains the published paper for DNAlive, and appendix B includes a comprehensive description of the DNA physical properties included in DNAlive, with a reference to their original publications.

Chapter 2

Biology

DNA was first isolated by the Swiss physician Friedrich Miescher who, in 1869, discovered a microscopic substance in the pus of discarded surgical bandages. Since then, much effort has been put in the investigation of this element and its role in the living beings. It was not until 1953 when, based on X-ray diffraction images taken by Rosalind Franklin and the information that the bases were paired, James D. Watson and Francis Crick suggested what is now accepted as the first accurate model of DNA structure in the journal Nature [CW53].

This chapter defines the terms that will be used in the rest of the Thesis, and provides a basic but detailed introduction to the concepts that are needed to understand this work.

2.1 Central dogma of the molecular biology

Living organisms can be divided between **prokaryotes** and **eukaryotes**.

Prokaryote cells lack a nucleus, cell membrane, and many other structures that are seen in eukaryotic cells; in general, they are less evolved and simpler. Eukaryotic cells are present in higher animals and their structure is very complex; a comparison of both cells is pictured in fig. 2.1.

Prokaryotes also contain genetic material —**DNA**—, but it is not packed (more on this in section 2.5) and its behavior is slightly different than eukaryotes.

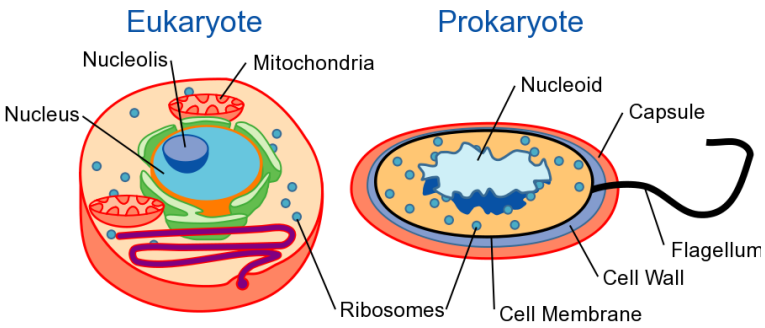


Figure 2.1: Simplified models for the cells of eukaryotes and prokaryotes.

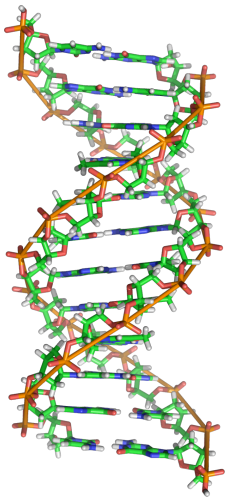


Figure 2.2: A chunk of DNA in its stable 3D form (Source: Wikipedia)

For this Thesis, unless specified, all references to cells are meant to be for eukaryotes, the ones present in human organisms. As stated, every **cell** of the human body contains a **nucleus**, where the DNA is surrounded by a liquid composed of water and other molecules.

It is well known that the DNA is a **double helix** (see fig. 2.2) that contains genetic information. However, this big molecule has very intricate physical and chemical structure, and plays a critical role in genetics.

An easy metaphor to understand the DNA structure is to view it as a long string, composed of character sequences. Each of the four letters that form the DNA `(A, C, G, T)` represent a small molecule that has a name, with its atoms, its bonds and its physical forces. These letters are named Adenine, Cytosine, Guanine and Thymine, and their generic name is **base**. Each one has a sugar molecule which acts as a scaffold, the **backbone**. The combination of a base with a sugar forms a **nucleotide**[1].

A chunk of DNA expressed as `GGCAATTACGACGGTATAACT` means that there are two Guanine molecules, followed by a Cytosine, followed by two Adenines, two Thymines, and so on.

Furthermore, every nucleotide is facing another nucleotide, forming the aforementioned double helix. The nucleotides always pair themselves in the following fashion: `C` pairs with `G`, while `A` pairs with `T` (or `U`, in RNA), and vice versa. For example, `AACT` always faces `TTGA`.

By convention, one of the **strands** is read from the top to the bottom, while the other is meant to be read from the bottom to the top. The "top" end is called the **5' end**, and the "bottom" is called **3' end**. A picture of the chemical representation of the DNA can be seen in figure 2.3.

Today, it is known that the DNA encodes the information that expresses all the human body. It has information on the hair color, the gender, the number

[1]However, in this thesis, *base* and *nucleotide* are used as synonyms.

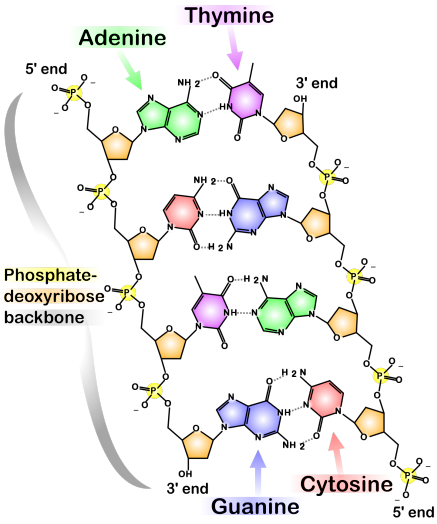


Figure 2.3: The DNA chemical structure. The left strand is read from top (5' end) to bottom (3' end), while the right strand is considered to be inverted. As presented here, each nucleotide is composed of different atoms, and the structure is held by atomic bonds. The "backbone" is the scaffold which subjects the nucleotides. (Source: Wikipedia)

of toes per feet, and so on. Many things about the DNA have already been discovered, like the translation process, but some others still remain unknown, like the gene regulation.

The **central dogma of the molecular biology** [Cri70] enunciates the normal flow of information that regulates genetics on living beings. Basically, it states that the DNA holds the genetic information, and the different means of transferring it. This flow can be interpreted as a graph, as depicted in Figure 2.4.

Each DNA sequence encodes some function in the body. One of the most important, however, is the production of **proteins**. Proteins are very complex molecules that, in turn, are formed by smaller molecules called **amino acids**. In every three DNA nucleotides the code for a protein molecule is expressed; for example, nine DNA nucleotides are equivalent to three amino acids.

In fact, there is another molecule involved in this process, the **RNA**, which is very similar to DNA but usually appears in a single strand form. It acts as a mold, being the intermediary between the DNA chain and the protein. Furthermore, in RNA, the nucleotide **uracil** (`U`) substitutes the DNA's thymine.

The $DNA \rightarrow Protein$ process consists of two phases:

- **Transcription**, where the DNA generates a piece of RNA.
- **Translation**, when the above RNA produces a protein. That protein then will express a gene.

The Central Dogma also explains a third information transfer: the **DNA replication**, used when the DNA duplicates, for example, when transferring

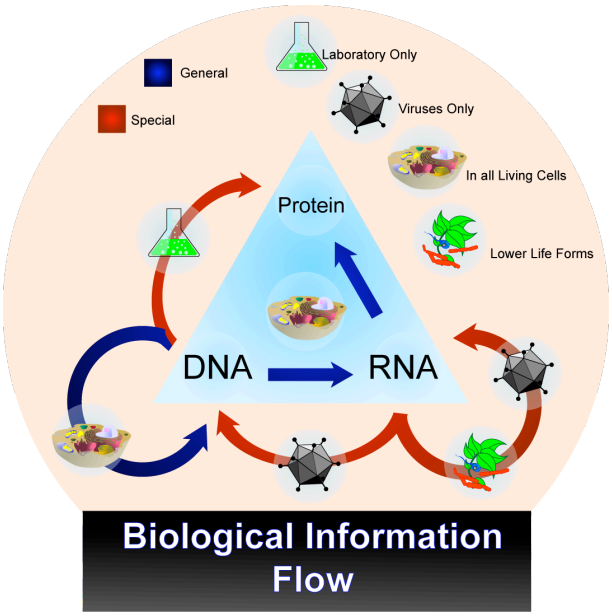


Figure 2.4: Central dogma of the molecular biology. The blue arrows determine the normal information flow, while the red arrows point to special cases. (Source: Wikipedia)

information to the progeny. In this three-way game (DNA, RNA, Proteins) there might also be other ways of transferring information, but they are very rare.

Even though the DNA/RNA express the Proteins, these latter molecules are essential parts of organisms and participate in every process within cells. They process food, build up the cells, and they even generate other DNA sequences. For example, the process of DNA duplication is carried out by a complex group of proteins that unwind the helix and, using another protein named **DNA Polymerase** , copy or replicate the master template itself. The proteins which regulate gene expression, binding to the DNA and activating the transcription process, are called **transcription factors**.

2.2 Genes

Genetics is the area of biology that studies **genes**, which is the basic unit of DNA information. All these are, in fact, DNA sequences placed strategically in the body cells. Using naive examples, one can say that, if the sequence `CCTTACAAAATAGGGTG` is present at about the 12,413,574th position of someone's 21st Chromosome, then that person is going to have green eyes.

In the last paragraph there were three remarkable pieces of information:

- the DNA sequence,
- the position and the Chromosome where it is placed (the **Transcription Start Site**, or TSS), and

- its function.

Today, many of the human genes rest still undiscovered, mostly because one —or more— of the above information is unknown. The most usual case is that scientists know that between the 12,400,000th and the 12,450,000th position of the human's 21st chromosome there is some sequence that regulates some unknown function, nothing else.

Nowadays there are very powerful computers that are capable to handle large character sequences and work with them. The whole Human Genome lengths about 3 Gigabases, three thousand million letters. But, how can we extract any data from just plain letters?

Given the fact that the DNA has physical properties that define its flexibility, and that a protein is not a language parser that reads the whole 3 Gigabases before attaching somewhere, the conclusion is clear: all interactions that happen in the human body and, in extension, everything regarding DNA and proteins, are ruled only by physical forces (e.g. electro-chemical forces).

Different families of genes are expressed by different proteins. For instance, **Polymerase II** (Pol-II), a very large protein with more than 10 subunits, is the main player in the transcription of genes encoding for **messenger RNA** (mRNA), the type of RNA that transports the information from the DNA to the proteins. However, it is also clear that Pol-II is not able to start transcription by itself, but needs of a large number of additional proteins.

Those proteins, named transcription factors (TF), create a large protein cluster bound to DNA, the **pre-initiation complex**, which precedes the transcription of the corresponding gene [SK03]. The region 200-300 bp[2] upstream the core promoter (see sec. 2.3), where the TF binds, defines the proximal promoter area, where multiple transcription factor binding sites are located. Additional signals are received from enhancers which can be bound far —even thousands of bases away— in the sequence of DNA, but that the DNA structure probably locates close enough as to allow interaction with the **pre-initiation complex**, the proteins involved in gene translation.

Genes can be divided in two big groups: **coding** and **non-coding**. The coding sequence determines what the gene produces, while the non-coding sequences are known to be genes, but somehow they do not seem to be translated. Coding genes account for about 30% of the total, while non-coding account for 70% of all genes.

2.3 Promoters

A **promoter** is a small region (hundreds of bases) located just before a gene, and it is the place where the transcription factors attach to start expressing a gene.

They are of extreme relevance because of several reasons. First of all, they are strictly related to genes, and finding a new promoter on a relatively deserted DNA sequence usually leads to find a new gene. Second, promoters regulate gene expression, which looks like Figure 2.5.

The **core promoter** is the region immediately upstream of the TSS, where the transcription initiation complex assembles. It is located upstream of the

[2]Base Pairs, a size unit meaning a base `A, C, G, T` and its complementary `T, G, C, A`

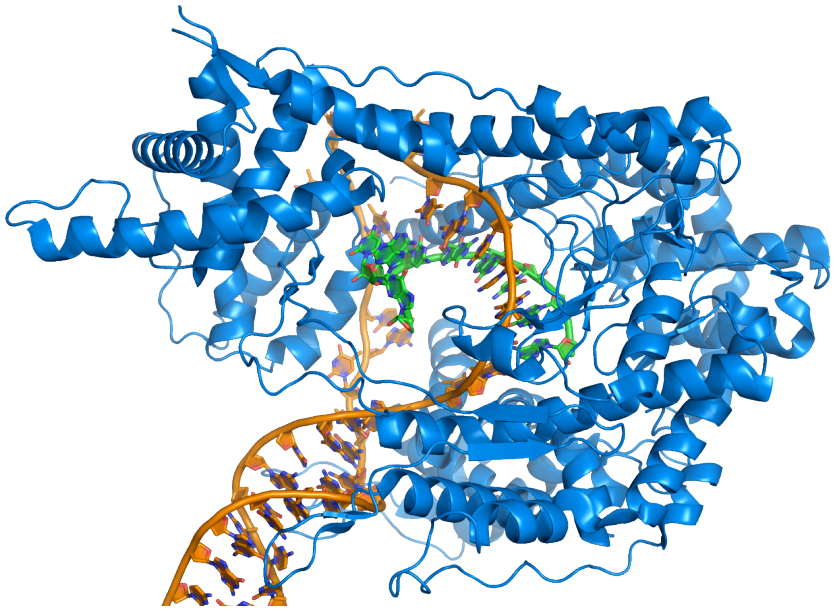


Figure 2.5: The process of gene expression. A Transcription Factor (blue, thin and curly) has bound to the DNA (orange, double strand) and is producing the RNA (green, in between the top opened DNA strands) which will, in turn, express the gene in the final Protein. (Source: Wikipedia)

coding part of the gene, sometimes up to several thousand base pairs, and is responsible for basal transcription as well as transcriptional regulation of the gene it is linked with.

Biologically speaking, there are many signals that help us detect promoters [AD07]. One is the **Initiator Element** (Inr), located around the TSS and can work independently or synergistically with the TATA box. However, the prevalence of the Inr element in mammalians is not clear, but it seems to be quite abundant in Drosophila[3]. TATA boxes and CpG islands are very clear signals that have been studied thoroughly, and they will be reviewed.

Still, the presence of a TATA box or a CpG island is not a requirement for the presence of a promoter, since there exist promoters without them. Similarly, their absence does not imply that there is no promoter.

2.3.1 TATA boxes

Early analysis of common eukaryotic signals at proximal 5' (see 2.3) upstream region of known genes revealed the presence of some over-represented motifs which were demonstrated to serve as signals for placement of the pre-initiation complex. The **TATA box** is the strongest of these signals and is recognized by the key TF IID complex. It has a consensus sequence `TATAAAA`, but large

[3]Drosophila is a species of fruit flies, used in laboratories because they breed fast, DNA recombination and mutations are common, and they share a lot of DNA with humans.

deviations from this consensus have been found in different genes [SK03].

It is generally found in the 25-30 upstream of the transcription start site (TSS), but again this distance can change depending on the organism [HS89]. Traditionally, the TATA box was supposed to be present in around 30% of genes, but it is not present in oncogenes, growth factors and house-keeping genes [SS03]. Furthermore, recent experiments performed within the ENCODE project[4] strongly suggest that the number ot TSS linked to TATA-box can be even lower [BTC+06].

2.3.2 CpG islands

DNA **methylation** is a type of chemical modification of DNA without changing the original sequence. It involves the addition of a methyl group (CH_3) to the DNA with the effect of reducing gene expression. DNA methylation at the 5th position of Cytosine has been found in every vertebrate examined [GGF87].

DNA methylation may impact the transcription of genes in two ways. First, the methylation of DNA may itself physically impede the binding of transcriptional proteins to the gene and secondly, and likely more importantly, methylated DNA may be bound by proteins that form compact, inactive chromatin.

CpG islands are unmethylated long (500-2000 bp) segments of DNA, with with at least 50% `C+G` content, and the number of CpG dinucleotides being at least 60% of what could be expected from the `C+G` content of the segment. The algorithm is further explained in section 5.2.1.

Promoters associated to CpG islands have multiple TSS that span a region of 100 bp or more [DGZ01] and usually lack other signals like TATA boxes, DPE or Inr elements [STS+01]. CpG island-associated promoters seem to be rapidly evolving in mammals, whereas TATA box promoters are more constrained.

Between 60-90% of all CpGs are methylated in mammals, and unmethylated CpGs are grouped in CpG islands. In many disease processes such as cancer, gene promoter CpG islands acquire abnormal hypermethylation, which results in heritable transcriptional silencing, avoiding the expression of some genes.

2.4 Physical properties of the DNA

The **mechanical properties of DNA**[5] are directly related to its structure. Every process which binds or reads DNA is able to use or modify the mechanical properties of DNA for purposes of recognition, packaging and modification. The extreme length, relative rigidity and helical structure of DNA has led to the evolution of techniques to compact a cell's DNA. The properties of DNA are closely related to its molecular structure and sequence, particularly the weakness of the hydrogen bonds and electronic interactions that hold strands of DNA together compared to the strength of the bonds within each strand.

One of the most important physical property of the DNA is its own shape. DNA appears in three conformations: **A-DNA**, **B-DNA** and **Z-DNA**, as

[4] ENCODE (the ENCyclopedia Of DNA Elements) is a public research consortium launched by the US National Human Genome Research Institute in September 2003. The goal is to find all functional elements in the human genome, one of the most critical projects after the successful completion of the Human Genome Project.

[5] In this document, the terms *DNA physical property*, *DNA descriptor* and *DNA mechanical properties* are used as synonyms.

depicted in figure 2.6. B-DNA is the standard form, and the *B* name is used only for description purposes, as it is commonly referred just as "DNA". A-DNA and Z-DNA differ significantly in their geometry and dimensions to B-DNA, although they still form helical structures.

The *A* form is likely to occur only in dehydrated samples of DNA, such as those used in crystallographic experiments, and possibly in hybrid pairings of DNA and RNA strands. It is a right-handed double helix fairly similar to the more common and well-known B-DNA form, but with a shorter, more compact helical structure.

Segments of DNA that cells have methylated (see 2.3.2) for regulatory purposes may adopt the *Z* geometry. It is a left-handed double helical structure in which the double helix winds to the left in a zig-zag pattern, instead of to the right, like the more common B-DNA form. There is also evidence of protein-DNA complexes forming Z-DNA structures.

Other conformations are possible; (C)ovalent mitomycin-DNA [SFL+95], (D)elta-DNA [SMHK01], (E)ccentric-DNA [VEH00], (L)ambda-DNA [SCHH82], (P)auling-DNA [ABLC98] and S-DNA [CLH+96] have been described so far, although they only appear in some specific organisms and are not common.

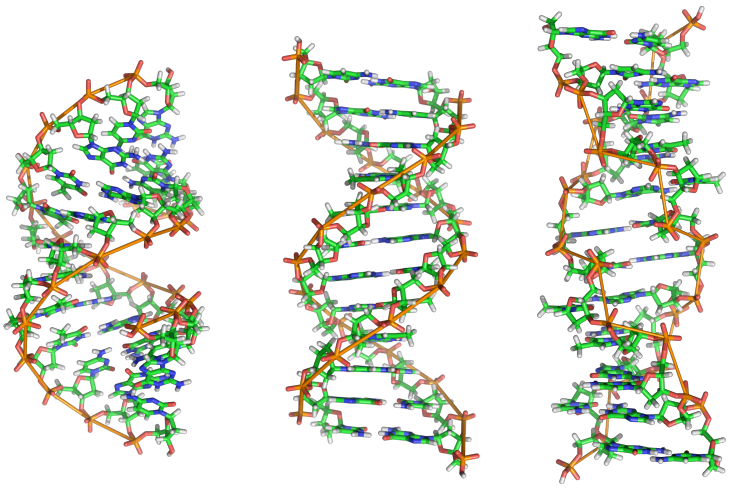


Figure 2.6: The structures of A-DNA, B-DNA and Z-DNA. B-DNA is the canonical form, discovered by Watson and Crick 50 years ago. A-DNA is typical on RNA, but also present in DNA sometimes. Notice how the Z-DNA winds to the left, whereas the A- and B-DNA forms wind to the right. (Source: Wikipedia)

At another structural level, we can also find DNA triplexes and quadruplexes. These DNA structures are composed of three or even four strands, instead of the typical two-stranded model.

The ability of the DNA to form these structures is known since the 50s-60s, but in the last years many studies have brought back the interest for them [SCH96], due to their possible implication in biological processes such as the transcription and to their potential biotechnological applications, like a viable information-encoding system. [LGK05]. Expanded DNA, especially quadru-

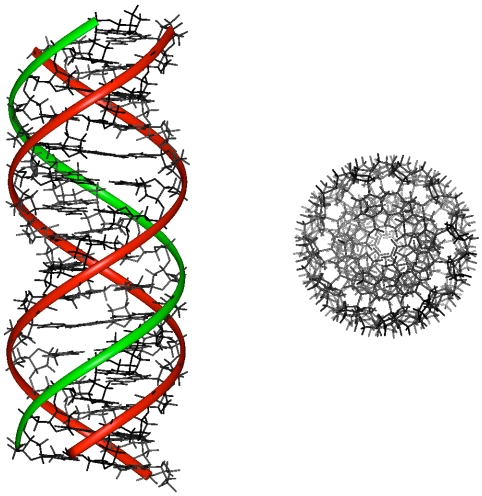


Figure 2.7: A triplex structure, side and top images. The two original strands are depicted in red, and the extra strand appears in green.

plexes, encode four bases of sequence information, and it forms antiparallel double helices of high stability and (generally) high selectivity.

A **triple helix** (fig. 2.7) is formed when a third string (RNA, DNA or a combination of both) is placed in the big hole of the double DNA helix and their bases interact with the hydrogen bonds. The triplex bases always form `(CG)C` or `(TA)T` triads.

The **quad-helix** (fig. 2.8, usually named quadruplexes or G-DNA, is an unusual DNA structure that is found at the bottom edges of the chromosomes, named **telomeres**. These regions are very rich in guanines, and are very important in the DNA replication because they allow the binding of the complementary strand. They are mostly present in the telomeric regions because they help to preserve and compact the DNA.

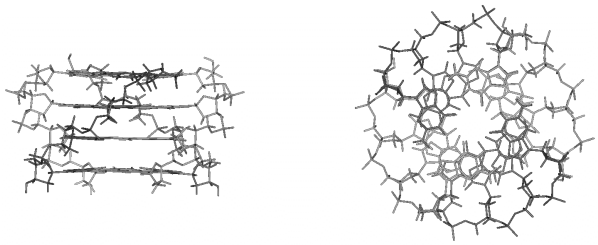


Figure 2.8: A quadruplex structure, side and top images. The whole structure is stabilized by a Na^+ or K^+ ion in the center.

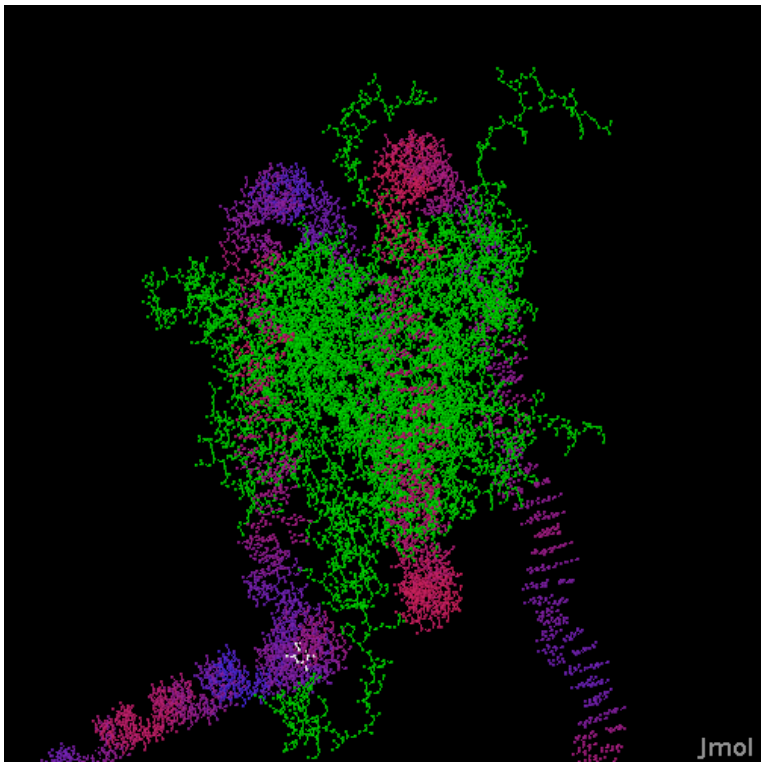


Figure 2.9: A nucleosome (green) bound to the DNA (red to blue scale).

2.5 Nucleosomes and Chromosomes

Human DNA is a very long string, about 3 billion bases in humans, divided into 24 chromosomes. As each base is 0.33 nm long, that makes about 1 meter of DNA in each human's cells; that is why DNA forms higher order structures.

The DNA is compacted many times with the help of proteins. The most basic one is the **Nucleosome**, a big structure that glues 146 base pairs in a loop shape, as can be seen in figure 2.9.

The DNA folding does not stop with nucleosomes. The resulting structure is compacted again and again, with the help of many different proteins, forming a substance named **chromatin** (literally: colored material, as it was discovered by staining the nucleus of the cells). The structural entity of the chromatin is the **chromosome**. The different compaction levels are depicted in fig. 2.10.

2.6 Concluding remarks

This chapter has presented a quick reference with all the definitions that are required to understand the biological background behind this Thesis. Further chapters reference some of this sections when needed.

The most relevant sections of this chapter are Promoters (2.3) and Physical properties(2.4). They define the basis for ProStar2 (ch. 5) and DNAlive (ch.

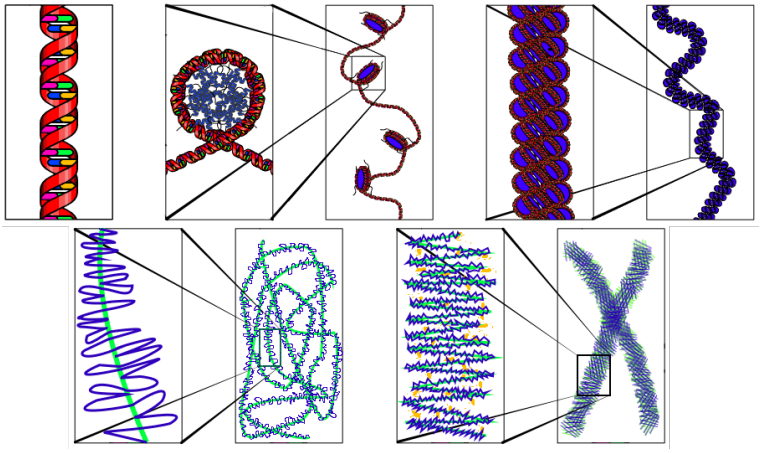


Figure 2.10: This picture shows how the DNA is compacted many times, forming chromosomes. The resulting complex, composed of DNA and proteins, is called chromatin. (Source: Wikipedia)

4), respectively, and more on these concepts will be explained in those chapters.

Next, we will overview some of the available tools which apply these concepts in bioinformatics software.

Chapter 3

State of the art

Bioinformatics covers a wide range of topics, which could be naively divided into databases and algorithms. Databases are especially extended and have become the most popular resource, but here only the most relevant are presented. Regarding DNA manipulation, the most accessed ones are Genome Browsers and BLAST-like software. Genome browsers allow the visualization of data in an intuitive manner, allowing the integration between databases. BLAST is a sequence searcher, and it has different versions for DNA, proteins and other structures.

Promoter prediction software is also abundant [BBB+06], however only recent methods have reached an acceptable level of accuracy. In this section two programs are analyzed, one being ProStar, and the other EP3, a recent work which shares some similarities with ProStar 2.

3.1 Genome browsers

As the genomic data grows faster and faster, researchers need tools to visualize the DNA sequences and their annotations. There is plenty of software aimed at displaying plain DNA structures, but the user needs to integrate further data by hand.

In 2002, a year after the human genome was fully sequenced, Kent et al. [KSF+02] developed the **UCSC Genome Browser**[1]. Its main objective is to provide a rapid and reliable display of any requested portion of the genome at any scale (fig. 3.1), together with several dozen aligned annotation tracks (databases). The browser also displays a huge amount of data related to the DNA, like RNAs, known genes, cross-species data and so on. Furthermore, users can add their own data tracks and display them along with UCSC's ones (see fig. 3.2).

Recently, the UCSC team has published another article with an update on the project [KKBB08]. The project is still alive, and there have been great improvements and many spin-off projects, like a genomic wiki.

[1] Publicly available at http://genome.ucsc.edu

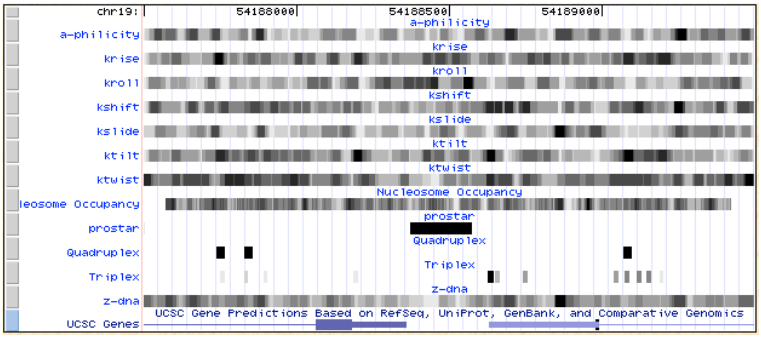


Figure 3.1: The UCSC Genome Browser plotting many DNAlive descriptors. On the top, there are the genomic coordinates; on the left, the name of each track.

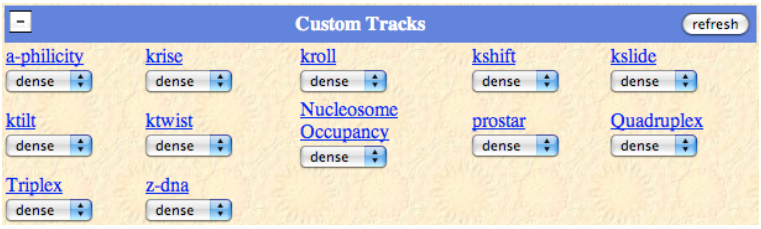


Figure 3.2: Managing tracks in the UCSC Genome Browser. The interface provides many alternatives to display the data, allowing a full customization of each track.

3.2 Sequence search

BLAST (Basic Local Alignment Search Tool) is an algorithm for comparing primary biological sequence information, such as the aminoacid sequences of different proteins or the nucleotides of DNA sequences. A BLAST search enables a researcher to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence. For example, following the discovery of a previously unknown gene in the mouse, a scientist will typically perform a BLAST search of the human genome to see if humans carry a similar gene.

BLAT is a BLAST-like tool designed to quickly find DNA sequences of 95% and greater similarity of length 25 bases or more. It keeps only an index of the entire genome in memory, which takes up a bit less than a gigabyte of RAM. The genome itself is not kept in memory, allowing BLAT to deliver high performance on a reasonably priced box. Then, a user can input a DNA sequence, and the BLAT server will quickly (5-6 seconds) return the genomic coordinates where that sequence is located; that is very useful to display annotated DNA from a sequence which was thought to be unknown.

3.3 DNA dynamics

Molecular dynamics (MD) is a form of computer simulation in which atoms and molecules are allowed to interact for some of time under the laws of physics, allowing the view of the motion of the atoms. Molecular systems are composed of a huge number of elements, so it is impossible to find the properties of such complex systems analytically. MD simulation uses numerical methods to avoid this problem. Its laws and theories stem from mathematics, physics, and chemistry, and it employs algorithms from computer science and information theory. Today it is applied mostly in materials science and biomolecular simulation.

MD simulations require a lot of computational power. Nowadays, simulating the movement of an average protein (10,000 atoms) for 1,000 nanoseconds[2] in 128 CPUs of a machine like the Mare Nostrum can take as much as 6 months of time and occupy 10 TB of data. Most MD, however, are simulated for 20-100 ns, reducing these values to a more handleable time. Of course, after running the simulation, the results must be extracted and analyzed, which accounts for most of the time of the researcher.

The general algorithm ruling MD is simple to understand, but very difficult to implement. Basically, it iterates the following process through time (typically, 10^{-15} seconds):

- Get the forces for each atom and its acceleration
- Move each atom to its new position

To get the atomic forces, a very CPU-intensive operation is used to compute the force field or energy potential. Then, corrections need to be applied, to fix the natural vibrations of the atoms. In addition, most MD use an explicit solvent, that is, the atoms need to be surrounded by water molecules, which exponentially increases all the calculations.

[2]MD time is measured in nanoseconds

Because of the complexity and expensiveness of classic MD, a *simulation of the simulation* technique is performed to determine basic molecular dynamics with a **Monte Carlo** (MC) method.

The MC approach relies on the distribution of random numbers to approximate expensive physical or mathematical calculations. Instead of trying to reproduce the dynamics of a system, it generates states according to appropriate Boltzmann probabilities. It employs a Markov chain procedure in order to determine a new state for a system from a previous one. According to its stochastic nature, this new state is accepted at random. The avoidance of dynamics —in Markov processes, each state is independent from the previous one— restricts the method to studies of static quantities only, but the freedom to choose moves makes the method very flexible.

3.4 Gene predictors

NSCAN, a modified version of **TWINSCAN** [GB06], is a paradigm of program detecting promoters based on gene structure. The program uses generalized **Hidden Markov Models** to find normal signals of genes locating the 5' end and conserved regions upstream, guessing then the promoter position. However, most of the programs for promoter detection are based on the idea that there are subtle sequence signals associated to promoters.

Thus, many algorithms are trained to recognize basic signals like the TATA box and/or Inr, CpG islands or regions with a large population of targets for Transcription Factors. The location of CpG islands has focused special effort and several methods have been independently trained to locate CpG-positive and CpG-negative promoters. Several of these methods use compositional rules, at the trimer, pentamer or hexamer levels[3], or in most sophisticated versions Hidden or Iterative Markov chains trained against known promoters. Finally, some methods like **FirstEF** [Dav03] or **Dragon Promoter Finder** [BSC$^+$03], **Dragon Gene Start Finder** [BS03] or **promH** [SS03] take advantage of the predicted gene structure to help their sequence-trained models to locate promoters.

The diffuse character of sequence signals at promoters indicates that factors other than the specific hydrogen-bond interaction between nucleotides and binding proteins modulate the recognition of target DNA fragments. As suggested by others [KB05], one of these additional factors can be the physical properties of DNA, which can control the degree of accessibility of the target sequences to binding proteins through the modulation of chromatin structure, the transmission of enhancers or proximal promoter information to the core promoter region, or the formation of protein aggregates in the pre-initiation complex.

The fact that DNA at promoter regions displays different physical properties than the rest, specially near the TSS, has been clearly probed by in prokaryotes and with less clarity also in different eukaryotes [PBCB98]; [ONGR01]; [FSD$^+$05].

[3] i.e. grouping the bases in sets of three, five or six elements

3.5 ProStar 1

After this exhaustive review of the promoter detection techniques, the MMB group decided to implement a mechanism to derive all the known physical properties and then analyze possible differential properties of DNA in regulatory regions in vertebrates. As expected, parameters are often correlated and the informative content is not always easy to determine [OcLNR02]. Beyond the idea that these parameters with stronger signals near TSS are supposed to play a determinant role in regulatory mechanism, they searched for this reduced set of parameters that were supposed to be able to cooperatively predict a promoter region.

In ProStar's case, first of all, the six helical force constants were calculated by means of MD simulations [PMSS07], as depicted in figure 3.3. Four duplexes containing several replicas of every possible combination of two nucleotides were used, plus four extra sequences related to CpG islands and TATA boxes: `GCCTATAAACGCCTATAA`, `CTAGGTGGATGACTCATT`, `CACGGAACCGGTTCCGTG` and `GGCGCGCACCACGCGCGG`. The trajectories were manipulated to represent the deformability of a given step along the three rotations and the three translations.

The dataset was gathered from protein-coding genes annotated by the HAVANA group, following the EGASP workshop rules[BBB+06]. Then, the method was trained for promoter recognition with a collection of 500-nts[4] sequences that comprised intervals of 250 nucleotides upstream and 250 downstream of the training TSS set. For the negative set, random 500-nts sequences from transcribed regions were selected, making sure that they did not overlap with the positive set. For the recognition of the strand, the method was trained with a collection of DNA sequences that comprised 900-nts upstream and 900 downstream of the same TSS. The reverse complementary sequences were taken as the negative set.

Using the MD derived parameters, any DNA sequence of size n can be described as a 6-dimensional deformation vector $v = (twist, tilt, roll, shift, slide, rise)$. For a given deformation, the values associated to every dinuecleotide step in the sequence are summed and then divided by $n-1$. For example, the twist deformation score of sequence ACGC would be $(0.036[AC]+0.014[CG]+0.025[GC])/3 = 0.025$. The 6-dimensional vector of the same sequence would then $v(ACGT) = (0.025, 0.033, 0.022, 1.200, 2.547, 8.230)$ (see fig. 3.4)

To classify the input sequences into the promoter class (k_x) or the non-promoter class (k_y), the Mahalanobis distance is used. Computing the physical properties of every sequence of the dataset a set of vectors for every class is concluded (X for class k_x and Y for k_y). The **Mahalanobis distance** D_M between the set of vectors X and Y is defined in eq. 3.1:

$$D_M(X,Y) = (\mu_x - \mu_y)^t C^{-1} (\mu_x - \mu_y) \tag{3.1}$$

Where μ_x and μ_y are the average vectors of the sets X and Y and C^{-1} is the covariance matrix of $X \cup Y$. The decision function g of a specific 500-nts DNA sequence with a description vector s to a class k_i with $i =< x, y >$ is defined in equation 3.2.

[4]Read as "500 nucleotides long"

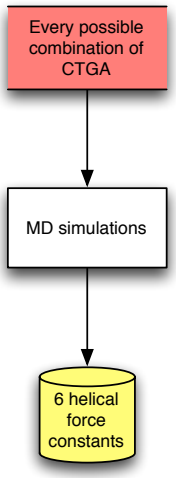


Figure 3.3: ProStar1: Calculus of the force constants

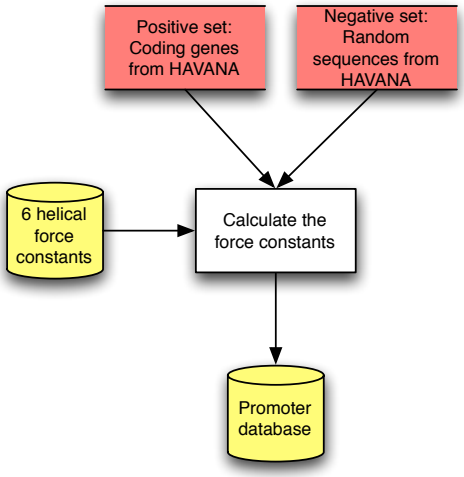


Figure 3.4: ProStar1: Generation of the promoter database

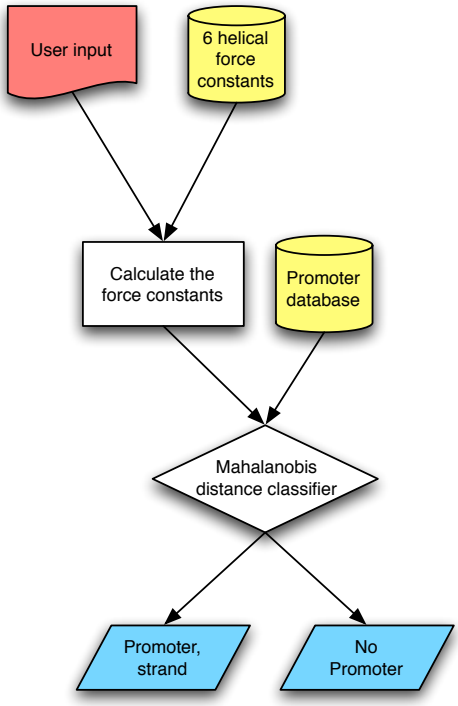


Figure 3.5: ProStar1: System architecture

$$g(s, k_i) = w_{k_i}^t s + w_{k_i,0} \tag{3.2}$$

Where $w_{k_i} = C^{-1}\mu_i$ and $w_{k_i,0} = -0.5\mu_i^t C^{-1}\mu_i$. When $g(s, k_x) > g(s, k_y)$, the sequence is classified as a promoter. Even so, the confidence of the decision can be modulated according to a normalized score defined in equation 3.3. If the score is greater than a specific threshold (set +1 by default), then the sequence is flagged as a promoter.

$$score(s) = \frac{g(s, k_x) - g(s, h_y)}{g(\mu_x, k_x) - g(\mu_x, k_y)} \tag{3.3}$$

The strand is predicted by recognizing the upstream/downstream signal asymmetry using a statistically discriminator based, again, on Mahalanobis metrics. Finally, very close clusters of predictions (using a 1000-nts window) are unified in a single hit. A graphical view of the system is depicted in fig. 3.5.

ProStar's method is conceptually and computationally simpler than any other general promoter prediction algorithm as it does not require any additional information, such as conservation of gene structure across species, presence of CpG islands, TATA-boxes, Inr elements or any other sequence specific signals. Due to its simplicity ProStar can be in principle applied even in cases where promoters are located in unusual genomic positions.

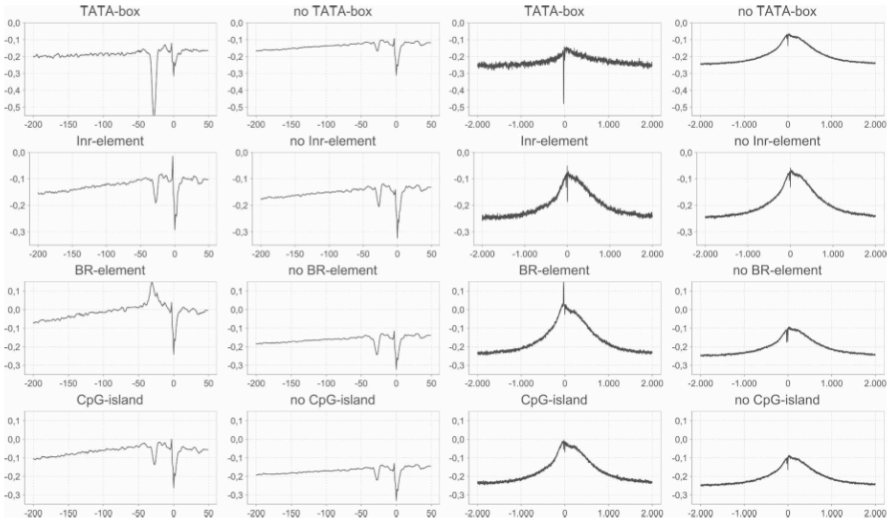


Figure 3.6: Physical properties study for EP3

The results of these comparisons show that despite its simplicity, ProStar behaved better than most of the other methods, similar to two algorithms that use gene structure for prediction (*fpom* and *firstef*) and only *nscan*, that is based also in multi-species homology, provides more accurate results for the reference set of genes. (see [GPTO07], pages 5–8)

3.6 EP3

Abeel et al. developed **EP3** [ASB+08], a novel approach for predicting promoters in whole-genome sequences by using large-scale structural properties of DNA. EP3 requires no training, is applicable to many eukaryotic genomes, and performs extremely well in comparison with the best available promoter prediction programs to the date of publishing. Moreover, it is fast, simple in design, has no size constraints, and the results are easily interpretable. Their method also has been tested on 12 additional eukaryotic genomes, including vertebrates, invertebrates, plants, fungi, and protists.

EP3 is very relevant for this Thesis because they analyze the relationship between different physical properties and the known promoter groups. Specifically, the variation of RNAP II is plot for each promoter descriptor: TATA boxes, CpG islands, TFIIB and Inr (see sec. 2.3). This graphic is depicted in figure 3.6

While EP3 does not outperform its peers by much, the program has several additional advantages compared with other promoter predictors. EP3 requires no training or parameter tuning, unlike other programs that need extensive amounts of experimentally determined data for the training of their model ([OSHN00]; [SKW00]; [DGZ01]; [DH02]; [BSC+03]). When working on a genomic scale, speed and memory requirements also are of importance. EP3 is very fast (for instance, it takes less than 1 h to annotate the complete human

genome), requires little memory, and can thus be run on a home computer; in contrast, some programs require a computer cluster of 80 machines for nearly a week to process the human genome. Besides performing very well, especially in light of its simplicity, EP3 can handle many eukaryotic genomes without modifications.

3.7 Concluding remarks

In this chapter we have described the most relevant works for gene prediction that currently exist. ProStar 1 is somehow the base for this thesis. EP3, by Abeel et al. [ASB+08], is relevant because it was published the same week than ProStar, and using a very similar methodology gets similar results, although for some specific promoter groups the accuracy varies between both programs. Also, its authors performed a study of the relationship between promoter signals and physical properties.

Chapter 4

Creating a platform for the analysis of DNA: DNAlive

In this chapter, the whole process for constructing DNAlive will be presented. A typical software engineering process was used since the beginning, composed of the following steps:

1. Define an objective
2. Gather the requirements
3. Design the architecture
4. Design the system
5. Implement the software
6. Test the software

4.1 Objective and requirements

The main objective of **DNAlive** is to integrate in one public platform all the descriptive information on the human genome with the physical properties of the DNA. In section 3.1 we presented the UCSC Genome Browser, the most popular genomic annotation tool, and we wanted to use it to display the information generated by the physical properties predictors.

Furthermore, we also decided to implement a 3D and 4D (animated 3D) viewer, because 2D maps sometimes are not visual enough; also, the platform needed the ability to search in databases for proteins which bind to a DNA and physically compute this binding to the double helix.

In general, the software needs to be able to to run many prediction and structural algorithms for a given DNA sequence, everything automatic, so that the user can get many data without being required to input further information. Data integration and calculus automation are the two main pillars of DNAlive.

The system should also have two main interfaces: a web page and web services. The first one should be compatible with the most used version of

every major internet browser. The web services must be compatible with other services and exchange data seamlessly.

The web page must be easy to use, work in real time and provide enough feedback so that the user always knows what to do next. For that, the performance should be tweaked to the maximum for all the data to be calculated in a few seconds. When the system finds that some calculus is going to take more than a minute, the user should be warned and asked for confirmation[1].

Regarding reliability, even though we will ensure that no data is lost, it is not a main requirement, as all the data can be regenerated and all the steps can be reproduced in a few seconds. Security is also important for the server part, and we must ensure that the software has no exploits. For the client part, however, no critical or confidential data is handled, so no extra security measures (SSL, passwords) need to be taken.

Finally, one of the most important requirements was modularity. As we needed to construct two interfaces for accessing DNAlive, every operation must be modularized.

4.2 System architecture and design

With modularity in mind, we generated a list with all the operations that we needed to implement and we built up an architecture diagram (figure 4.1)

The user only needs to input genomic coordinates or a DNA sequence. They are interchangeable in most cases, as the software can use BLAT [Ken02] to look for the coordinates for a given sequence or retrieve a chunk of DNA for a specific genome.

Once the input data has been processed, the user can calculate the physical properties for the sequence and plot a 2D map of them. If these coordinates belong to a genome, the UCSC genome browser can be used for the plot. Else, we need to implement our own 2D plotter.

Then, we generate the DNA 3D structure from the sequence, calculate its physical properties, and plot the 3D or 4D images. Bound proteins can also be searched for, as an optional step. We will run an algorithm that looks for proteins on a database given a DNA sequence, and will return the results. After that, another algorithm will try to bind the proteins to the DNA sequence, and then return a DNA-Protein structure with the best matches.

It should be noted that, for the 4D plot, no physical properties are computed. This was a design decision, derived from software constraints. When we tested the animation software we discovered that painting physical properties on a 4D animation was very resource-demanding and the system went unresponsive, which contradicts the philosophy of DNAlive: the web must feel real-time interactive and fast to the user.

4.3 The physical properties scripts

For a gene to express *(e.g. Blue eyes)*, a protein *(Transcription factor)* attaches to the DNA sequence associated with that gene `(CATGACTGACGTACGTTAGC)`, and

[1] In computational biology, a computing time of less than one minute is very fast.

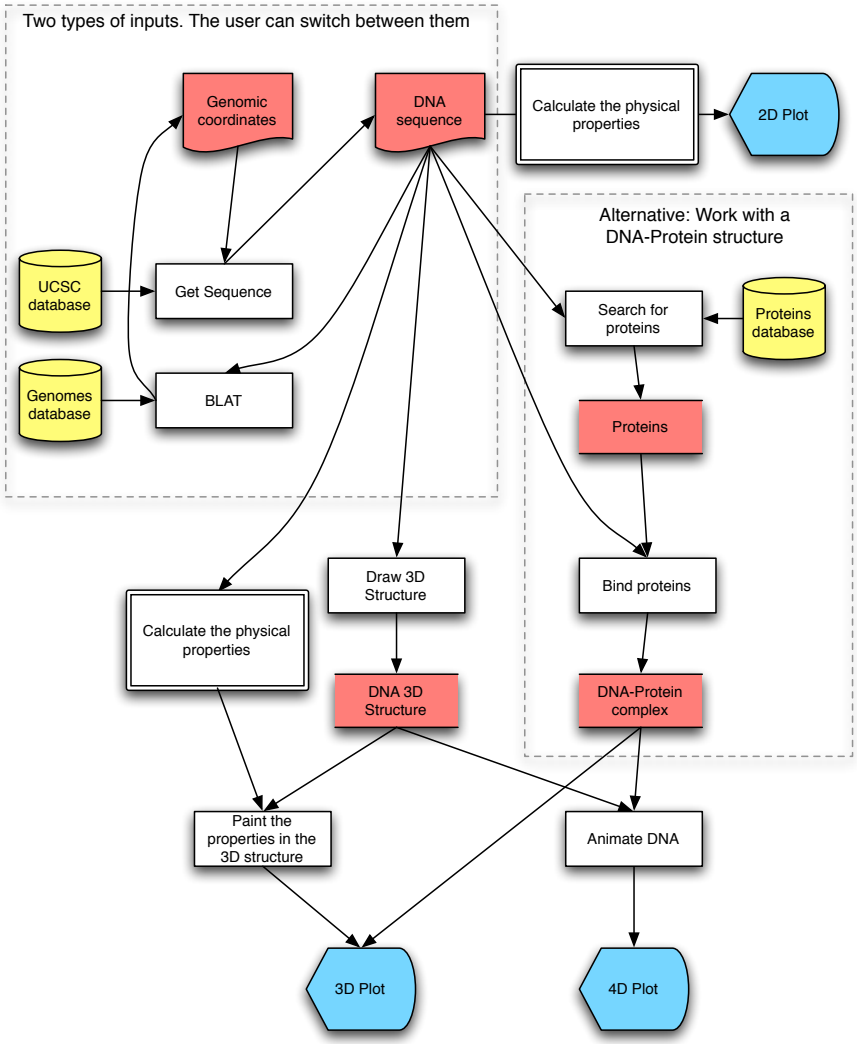


Figure 4.1: Architecture of the system. The user can input either genomic coordinates or a DNA sequence, while being able to change between them at any moment. The outputs (2D, 3D, 4D plot) are shown in blue, and all the data structures are painted in red. Databases are painted in yellow, and all the other processes are depicted in white.

then produces another protein. These proteins do not actually bind to a specific DNA sequence, but to a sequence with specific **physical properties** or, computationally speaking, its attributes.

In Chapter 3 we presented a list of different works on DNA descriptors. For this phase we chose 29 of them, described in the tables in appendix B.

Some examples of physical properties are:

- The melting temperature. Some parts of the DNA are more easily melted than others; just like the edge of a paper burns more easily than the middle part.
- Twist. The DNA can be physically twisted more easily in some specific areas.
- Nucleosome preference. The probability that this DNA segment will bind to a big protein named Nucleosome (see section 2.5)

	AT	TA	GC	TT	GG	CA	AC
Melting temperature	0.023	0.102	0.073	0.009	0.069	0.104	0.044
Twist	0.033	0.017	0.025	0.026	0.026	0.016	0.036
Nucleosome preference	0.012	0.038	0.081	0.011	0.012	0.021	0.054

Table 4.1: Sample values for different physical properties

So, how can these properties be calculated? There are various methods, but the main idea is to compute the value for each couple of nucleotides, and then average them if needed. In table 4.1 we can see how the DNA "alphabetical" sequence can be translated to plain numbers.

Let us imagine that we have the following DNA sequence:

`CGTTAATATCGGCGTAGCTAGTGTTTTTCCGATATATCAGTCCGGGCCGCG`

That sequence can be interpreted as follows:

`XXXPPPPPPPPPPXXXXGGGGGGGGGGGGGGGGGGGGGGGGGGGGXXXX`

being `P` a promoter, `G` a gene, and `X` nucleotides with no useful information.

We can transform this sequence to plain numeric values (again, see table 4.1) and, if needed, plot a histogram of the resulting values, as shown in figure 4.2

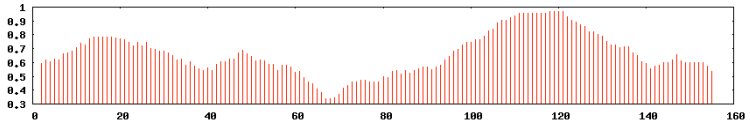


Figure 4.2: Plot of the "A-Philicity" property for a given sequence

Most of the descriptors use the Nearest Neighbor (NN) model. This approach assumes that the stability of a given base pair depends on the identity and orientation of neighboring base pairs.

The energy values are retrieved experimentally, in vitro. Scientists calculate the energy values for 500 DNA sequences [BD98] and, then, using a linear equation system, the values for each couple of base pairs is determined.

Once the energy values for each couple of base pairs and each property is calculated, the algorithm needs to read the DNA sequence, aggregate the values every couple of base pairs, and apply some correction factors. Usually, a sliding window of 25 base pairs is used, so that every 25 bases the value is averaged and the signal noise is reduced. These algorithms are, in fact, physical properties predictors, as they can only estimate the actual value of the descriptors with a mathematical model which is not perfect. However, this approach is widely used and drops results which are very similar to the experimental ones.

To implement the scripts we used the Perl programming language. The main reason is that there are already plenty of libraries for bioinformatic use. It also interacts easily with web pages, and each script could be transformed into a web page if needed.

4.4 Web page

The web page should be the main interface for DNAlive, as many users know how to use a web browser. It needs to be easy to use, provide many help links and guide the user through the process.

The technology choice was clear: HTML+CSS with Javascript for the page, and PHP for the server-side scripts. Our server uses Apache with PHP5, which is a very stable environment and allows object oriented programming. Our bad experiences with Java web servers (see section 4.5.2) made us discard the possibility of using JSP pages.

A PHP page was implemented for every major function, using Javascript and hidden frames to exchange information between the pages. The main page would call the helper pages when needed, retrieve the information, and call the next subprocess. For the scripts side, we used temporal folders on the server for exchanging information. A random token is generated every time that the user reloads the page, allowing the server to centralize all the data on a single folder and also providing a cache mechanism.

For the design details, the sample CSS from the MMB group pages (`mmb.pcb.ub.es`) was used, with slight changes to improve readability. The *Scriptaculous*[2] Javascript library helped on the page transitions and provided a beautiful and professional touch to the user interface.

Based on similar bioinformatics pages, we initially designed a very long form where the user could input a lot of information. Unfortunately, the interface was crowded with options and it was changed to a step-by-step guided process (more in sec. 4.5.2). We then took advantage of the clean, new code and designed a logging system and a development-production dual system based on best-practices advices from [Hen06].

The logging system keeps track of any unexpected outputs or errors and —if the error is fatal— redirects the user to an informative page announcing that we are aware of the problem, and then we get a notification. This way, we could track many hidden bugs and solve strange problems.

The development-production system keeps always an immutable production version, where the beta testers discover new errors on a frozen environment, while allowing the developers to work on new features and fix existing bugs. The testers can then be notified when there is a new release version and check again

[2] http://script.aculo.us/

if the bugs where fixed and provide feedback on the interface changes. This system also keeps backups of previous versions and deploys the development page to production instantly, so no downtime is noticed.

4.5 Web services

The *Instituto Nacional de Bioinformática* is one of the main providers for **BioMOBY web services**[3]. The services are scattered through the many INB nodes, but the repository is hosted at the INB server.

BioMOBY [WL02] is an open source research project which aims to generate an architecture for the discovery and distribution of biological data through web services; data and services are decentralized, but the availability of these resources, and the instructions for interacting with them, are registered in a central location called MOBY Central. BioMOBY adds to the web services paradigm, as exemplified by Universal Data Discovery and Integration (UDDI), by having an object-driven registry query system with object and service ontologies. This allows users to traverse expansive and disparate data sets where each possible next step is presented based on the data object currently in-hand. Moreover, a path from the current data object to a desired final data object could be automatically discovered using the registry. Native BioMOBY objects are lightweight XML, and make up both the query and the response of a Simple Object Access Protocol (SOAP) transaction.

After analyzing the features of the system, we decided to implement all the original algorithms from the web page into BioMOBY web services. We also provide other services so that a complete workflow can be launched. **Workflows** are nothing more than web services whose inputs and outputs are connected and can generate many different outputs from just one input.

BioMOBY web services require very specific software versions and libraries. They need to be implemented in Java 1.5 and jBoss 4.0.3sp1, as the most recent versions of these software have changed the XML-RPC APIs and BioMOBY does not support them yet. Nevertheless, it is an open platform, and there is already some people at the INB working on a new API. The current one, however, can be obtained via SVN, and then compiled and deployed to the jBoss application server.

To develop new web services we used **Taverna** [KSW06], a tool that automates the process. First of all, the new service needs to be registered in the yellow pages (in our case, the INB server), including all the input and output data structures, the description, and so on. After that, Taverna generates the skeleton for the web service, which needs to be imported into the development IDE of choice —we used Netbeans—; then, it can be programmed as a standard web service. The skeleton and the endpoint call the API when needed but, in practice, we needed to perform some tweaking of the generated skeletons.

Two sets of web services were implemented: a workflow to calculate all physical properties for a given DNA sequence (fig. 4.3), and another workflow to generate annotations on a sequence (fig. 4.4).

[3] http://www.inab.org/MOWServ/

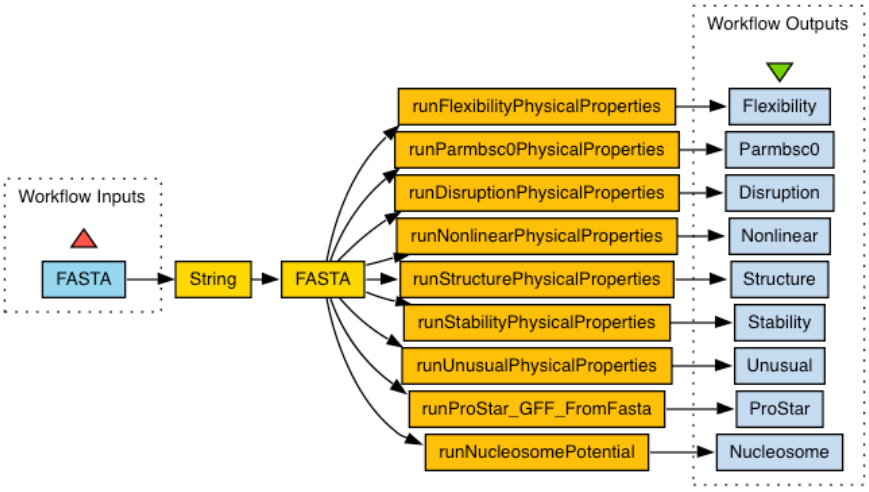


Figure 4.3: Physical properties web services

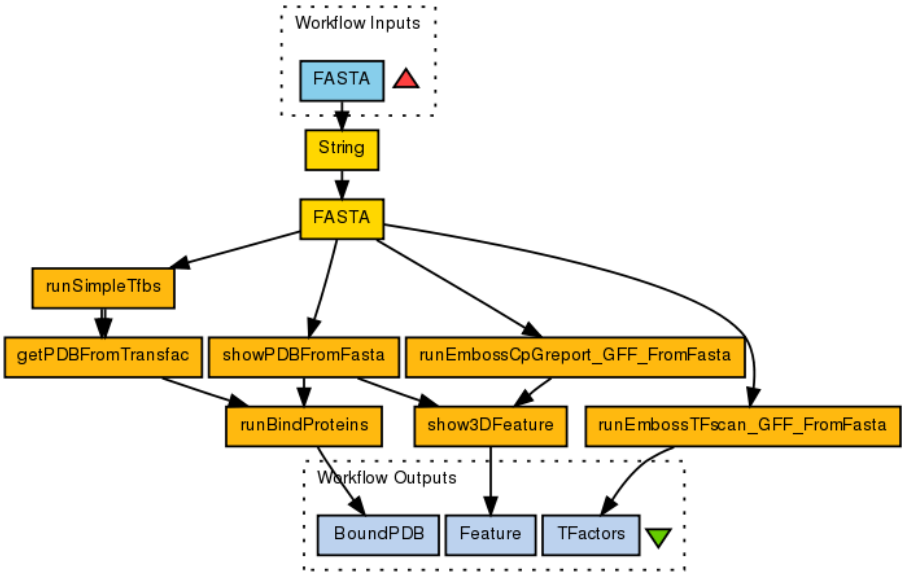


Figure 4.4: Structural web services

4.5.1 Physical properties services

There is no need to dig deep into the implementation of these services, because they are only a wrapper for the scripts. They receive the input data in the standard BioMOBY format, call the scripts, reformat the output into BioMOBY data, and return it to the user.

The services are grouped by descriptor types: DNA Flexibility, PARMBSC0 helical force constants, DNA Disruption energy, DNA non-linear dynamics, DNA 3DNA structure, DNA stability, Unusual DNA conformation, ProStar and Nucleosome potential[4].

The user can build a simple workflow, like the one in fig. 4.3, and launch batch calculations for different DNA sequences, automating the process.

4.5.2 Structural services

The structural services allow the user to perform intermediate calculations or just use our algorithms without an internet browser. The equivalences between the architecture of the system (fig. 4.1) and the structural services workflow (fig. 4.4) are:

- The service "showPDBFromFasta" converts a DNA sequence into a 3D structure, in the PDB format[5]. This implements the use case "Draw 3D structure"
- The DNA-Protein use case (architecture: "Search for proteins", "Bind proteins") is carried out by "runSimpleTfbs", which searches for proteins; "getPDBFromTransfac", which retrieves the proteins from the database; and "runBindProteins", which finally binds the proteins to the DNA.
- There is one new use case: "show3DFeature", whose inputs are a DNA 3D structure and information about the CpG islands ("runEmbossCpGReport_GFF_FromFasta"). This service outputs the DNA and a script to draw the CpG information in the VMD software. The user then needs to open this software, open the DNA molecule and paste the script code.
- Another service is provided: "runEmbossTFscan_GFF_FromFasta", which outputs information about the transcription factors on a DNA. It is a wrapper for the program `tfscan` from EMBOSS, a bioinformatics software package. The results are similar to those of "runSimpleTfbs", and is offered as an alternative to it.

Looking again at the system architecture, there are some process which have no equivalent as a web service. First of all, we decided not to implement those which are already provided as a web service by somebody else. In this case, "BLAT" and "Get Sequence".

Animating a DNA is also a web-only feature. We think that it is important to offer services which can be useful to run as a batch, but watching a DNA movie is an interactive process and it makes no sense to return a long file. With

[4]In the web page, these two services appear as "Regulation parameters", but we decided to separate them to publicize ProStar, letting it appear as a single service.

[5]PDB is the de-facto standard format for 3D DNA structures, and FASTA is a widely extended format used to store sequences.

3D DNA it makes sense to return a PDB file, because they are widely used and there is plenty of software to derive data from them.

4.6 Deployment and testing

Both interfaces were deployed on `mmb2.pcb.ub.es`, the MMB group's application server.

The web page stayed on a private beta phase for about five months, and since very early stages many of the group members tested it frequently, thanks to the dual development-production release cycle and the log system. The tests were also very important for determining the final interface of the web, as the first versions were hardly usable.

The implementation cycle for a web page is very interesting, as the testing part accounts for most of the development time. It is composed of many modules, written in different languages, and bugs are difficult to track. There is also the security problem, because a faulty input can open a breach in the server, and all the parameters must be checked at least once in every layer (Javascript, PHP, Apache).

A problem arose from one of the external software we use: Jmol, the 3D molecular viewer, has very severe memory limitations. We investigated further and discovered that Java Virtual Machines run very limited on memory. They have a heap size of 64 megabytes for security reasons, because Java applets run in client mode and could hang the system if the software is badly programmed. That limits the size of the molecule that we can display and, furthermore, all operations on the molecule cost a lot of CPU time, so we contacted the Jmol team and found some workarounds to this issue. Still, at the time of writing this document, the input for the modules which use Jmol is constrained, and there is nothing that we can do.

Still another testing needs to be done, that is, browser compatibility. Our group works with both Linux/Firefox and Mac/Safari machines, but Windows/IE is still widely extended among researchers —and, more important, research group leaders—. After much tweaking, we support IE6 or higher, Firefox2 or higher, and Safari2 or higher, on all the platforms that run these browsers. Opera 9+ also works, but it is not officially supported.

The web services were not tested exhaustively, as they do not really allow the user much flexibility. If the input is valid, then the output will be correct. Else, no output will be generated. Unfortunately, the 4.0.3sp1 version of jBoss is not maintained anymore and has many bugs, so the main problems came from the application server. Once we fixed most of them, the web services stay stable and work fine.

To date, both the page and the web services have been working without an issue.

4.7 Concluding remarks

The development of a large web application like DNAlive requires not only web developing skills but also good planning and following an engineering process.

The web page needs to exchange data with external sites, thus using their available APIs.

Layered software design is essential in this case, since the data access and the system logic need to be independent from the interface. Changing the algorithm to compute a physical property, for example, cannot require modifications in the interface code.

Finally, to develop research tools which can be computationally expensive, one must also analyze the available hardware. Some of the scripts require exponential algorithms, so all the batches need to be run through a queuing system, to ensure hardware stability.

Chapter 5

Applying AI techniques to ProStar

While Phase 1 was mainly devoted to the development of the DNAlive environment, in this chapter we will present the new methodology used to improve ProStar, whose original version was presented in section 3.5.

In order to follow the EGASP rules [BBB+06] and to be able to directly compare the results with those of ProStar1, we got the same list of positive promoter sequences and the set of non-promoter sequences as in the previous version.

Positive sequences are the collection of 885 genes, annotated by the Havana group in the ENCODE region (1% of the human genome) belonging to coding genes. This information is stored on an index file `havana.gene`.

`havana.gene` contains the name of the gene, the name of the ENCODE region where it belongs, the strand, the Transcription Start Site (TSS) and the end of the transcribed region. A sample follows:

```
Havana1 ENm001 + 354319 390710
```

Considering the TSS as position 0, we selected the 2000 bases around it $[-1000..+1000]$ on the ENCODE database, resulting on 885 sequences, 2000 base pairs each. Further data calculations required us to drop 11 elements, remaining 874 of them.

For the negative set, we got sequences composed of 2000 non-overlapping bases (to genes and to themselves) lying in the same strand of transcribed regions of the Havana genes. Using the example above, where 354319 is the TSS, we retrieved the bases 353319 to 355318 for the positive set, and then, in groups of 2000 bases, the rest of the data from 355319 to 390710, is added to the negative set. Later on, we refined the negative set, deleting the elements which were too close to a gene, as some of their properties were too similar to those of the genes.

The strand information is also important. As the ENCODE database only contains the positive strand, if the gene is located in the negative strand, the resulting sequence needs to be complementary reversed. This means that `AAGT` is reversed to `TGAA` and then complemented to `ACTT`.

5.1 Proposed solution

We are presented a classification problem which cannot be described by any existent model. So, a new model needs to be discovered, and we propose to learn it by using a neural network.

There are 29 DNA descriptors which we believe can be used to build this model. However, we cannot use all of them to train the network, so we will apply some filters beforehand, to the data set and also to the descriptors.

First, we will split the promoter database in four groups to gain specificity. Then, an analysis of the DNA physical properties will be performed to reduce the input noise on the network (fig. 5.2). After these techniques have been applied, we will train a network for each group and study its performance as a promoter classifier (fig. 5.3).

5.2 Promoter division in four groups

Promoters were introduced in section 2.3, along with their two key features: TATA boxes and CpG islands. It has been widely proved [LGLP92] [SK03] [MVDC+08] that these key elements, if present, affect the overall behavior of the promoter.

Biologically, it is appropriate to distinguish those elements *a priori*, because grouping promoters based on the presence of TATA/CpG can lead to more accurate predictions.

A **TATA-box related promoter** is defined [Buc90] when, for a given 2000-nts DNA sequence, with the annotated TSS in position +1000, there is at least a 20-nts subsequence in the 9900-1000 range with a TATA-Score greater than a cutoff value.

This **TATA-Score** is defined as the sum of the corresponding values for the TATA weighted matrix, the most important part of which is depicted below. To interpret it, read the DNA sequence in windows of n bases, being n the length of the TATA matrix. Then, match each base to the corresponding value, and sum all of them. If that sum is greater than the cutoff value, usually fixed at -8.16, a TATA box is found.

As an example, the sequence `GTATATAAG` will have a TATA-box ($0-0-0-0-0-0.52-0-0-0 = -0.52 > -8.16$), while the sequence `AACCGGTTA` will not have one ($-1.02-3.05-5.22-3.49-3.77-4.73-3.65-0.37-0 = -25.3 < -8.16$).

Notice that the TATA box is very flexible in its sequence, as the example `CTTTGAATG`, which one might not label as a TATA box at first sight, has a TATA-score greater than the cutoff ($-0.28-0-2.28-0-3.77-0-0-0.37-0 = -6.7 > -8.16$)

For a given 2000-nts DNA sequence, with the annotated TSS placed in the position +1000, it will be considered associated to a **CpG island** if it contains a region of greater than 200 bp with a percentage of G and C greater than 50% (eq. 5.1), and the observed/expected ratio of CpG is greater than 0.6 (eq. 5.2) [GGF87]. CpG refers to a C nucleotide immediately followed by a G. The "p" in "CpG" refers to the phosphate group linking the two bases.

$$\frac{\#C + \#G}{window\ length} \tag{5.1}$$

Pos.	-3	-2	-1	0	1	2	3	4	5
A	-1.02	-3.05	0	-4.61	**0**	0	**0**	**0**	-0.01
C	-0.28	-2.06	-5.22	-3.49	-5.17	-4.63	-4.12	-3.74	-1.13
G	0	-2.74	-4.28	-4.61	-3.77	-4.73	-2.65	-1.5	0
T	-1.68	0	-2.28	**0**	-2.34	**-0.52**	-3.65	-0.37	-1.4

Table 5.1: Partial weight matrix for the TATA box, the full version can be found at [Buc90]. Values for the "TATAA" sequence are rendered in bold.

$$\frac{\#CpG \times window\ length}{\#C \times \#G} \tag{5.2}$$

5.2.1 TATA/CpG clustering

Both the positive and the negative set of sequences are clustered in 4 groups using the aforementioned methods. The plus and minus symbols indicate if the TATA box or CpG islands are present.

	Pos	Neg	Total	%Total
`TATA- CpG-`	407	4960	5367	75.6%
`TATA- CpG+`	377	288	665	9.4%
`TATA+ CpG-`	60	959	1019	14.3%
`TATA+ CpG+`	30	23	53	0.7%
Total	874	6230	7104	100.0%

Table 5.2: The dataset, divided into the four promoter groups.

This way, from now on, every cluster will have each its own positive and negative set and will be treated independently.

The EGASP rules establish that 13 of the 44 ENCODE regions are used for training and the others for testing. Unfortunately, after clustering the promoters in four groups, a further division between a training set and a test set would leave the smallest group empty, and the rest with few elements. Therefore it was decided to use the whole dataset for training, performing a 10-fold cross-validation.

5.3 DNA descriptor analysis

There are 27 physical descriptors of DNA to be included in the new version of the promoter prediction, 26 of them belong to the 29 descriptors in section 4.3. We believed that CpG islands are highly correlated with most physical properties, so an extra descriptor for CpG islands was added[1]. On the other side, the Triplex and Quadruplex algorithms are very experimental and we might get unexpected results, so they were discarded. ProStar1 was removed because it

[1]This is the same concept as in sec. 5.2.1. Now, instead of using the official threshold for deciding whether a CpG island is present, we will use all the resulting values as a descriptor.

is just a composition of the 6 helical properties, which are going to be analyzed again.

We then run the descriptors on all the dataset, positive and negative. The computing time took about 48 hours in 10 multi-core machines of the MMB grid system; an estimated time of 1000 CPU-hours. Even though the process could have been run through DNAlive web services, the naked scripts were used, so that the software could be run through batch queues, avoid network and web-services protocol overhead, and speed up the calculation time in general.

The descriptor analysis consists of two steps, depicted in figure 5.1. The first step is to compute the Pearson correlation matrix for all descriptors, in each promoter group. The results will show if there are significant differences between promoter groups.

Then, a Principal Components Analysis per group was performed. The role of the PCA is to gather the most important properties that define promoters, and reduce the weight of those which do not add important information.

5.3.1 DNA parameter correlation

For every promoter type, we tried to narrow the descriptors to avoid overtraining our system by using parameters that do not add more information to the promoter definition, so we studied the correlation of the properties.

The data now consists of 7104 elements (genes and non-genes, see table above), with 27 descriptors for each. The descriptors are standardized arrays of 2,000 values, one for each nucleotide of the sequence.

In order to handle this amount of data, we decided to keep just 40 elements for each array. The average for each 50 elements was calculated, non-overlapping, resulting in $2000/50 = 40$ elements per array. Now each element of the dataset has $27 * 40 = 1080$ descriptors, resulting in a more manageable input matrix I of 7.6M items, but it was needed to reduce it even further to feed the neural network (tab. 5.3).

Element	cpg_50	cpg_100	cpg_150	...	z-dna_2000
Havana1	-0.649241	0.028227	0.028227	...	0.705696
Havana5	-1.147745	-0.179814	-0.386870	...	-1.55970
...					

Table 5.3: Sample descriptors matrix I before applying the eigenvectors function.

The first task was to run a **Pearson correlation** between all descriptors:

- For every promoter, every parameter is computed in 50-nts non-overlapping windows, having 40 lectures per parameter per sequence.
- For every couple of physical properties, the Pearson correlation coefficient is computed, ending with a 27x27 symmetric matrix.
- Finally, the Pearson correlation is calculated for each combination of properties and each promoter group.

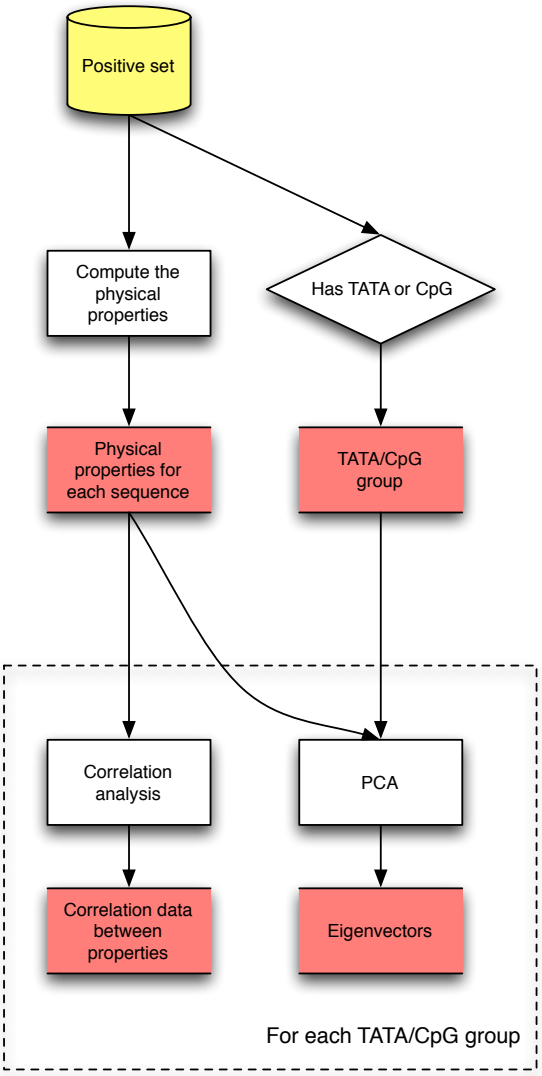


Figure 5.1: Descriptor analysis methodology.

In general, there should be correlation between some physical properties, either direct or inverse correlation. For example, the stability, by definition, is inversely correlated to the free energy (fig. 6.5).

The results (fig. 6.4), indeed, show that some of them reach correlation values even higher than 90% (colored in red in the aforementioned figures).

We decided then to run a PCA with all descriptors. As the algorithm builds a correlation matrix to extract the eigenvectors, it should ignore (i.e. assign low weights) to highly correlated descriptors and positions. Some of the blatantly correlated properties could have been removed by hand, but we decided to test the PCA analysis power by running it with all the data.

5.3.2 Principal components analysis

Pearson correlations are a useful first approach for analyzing data, but the resulting information (see Results, 6.2.2) was not enough to train a neural network. A **Principal Components Analysis** can automatically extract the most relevant data of a big matrix, reducing all the variables to a combination of eigenvectors.

Even though the negative set is about 7 times the size of the positive set, the groups will be balanced to run the PCA. The conservation of information was set to at least 80%.

Unfortunately, the initial results of the analysis showed a bias towards the sequence position, as the first eigenvectors of the PCA were defined by sequence positions and not the properties. For example, the position 1450 of the sequence was always present in the first eigenvector for all the properties, then position 800 in the second eigenvector, and so on. That indicates a lot of noise caused by the negative set and, while it is understandable and provides useful information on the relevance of the sequence position for both sets, it was not what we wanted.

Instead, we ran another PCA only on the positive set, getting better results. In that case, eigenvectors were defined by properties with different positions, which is more relevant for our problem.

The resulting eigenvectors look like $E_i = weight \times < property, position >$. Thus, a formula can be derived to transform the input data matrix $I =< property, position, value >$, into a vector R with as many elements as eigenvectors has that promoter group.

$$R = \sum_i I \times E_i \tag{5.3}$$

That way, we can transform the 1080 input parameters to only a few dozens (Results, table 6.1), while conserving 80% of the original information. With less parameters, the neural network to be trained (see next section) will be faster and it should not overfit.

5.4 Predictor training

Once the descriptor analysis was performed, the next step consisted in training a neural network to classify the input sequences into promoters and non-promoters. A neural network will be built for each promoter group, because their

signal for each physical property is different. Furthermore, the PCA reduced the number of inputs, weighting the most important descriptors and ignoring the others.

It is important to note that, even though we apply the positive set eigenvectors function to the input data, the neural network is fed with both sets to provide a better profiling of each class. They were balanced in the following manner:

	Pos	Neg	Total	%Total
`TATA- CpG-`	407	407	814	49.3%
`TATA- CpG+`	377	288	665	40.3%
`TATA+ CpG-`	60	60	120	7.2%
`TATA+ CpG+`	30	23	53	3.2%
Total	874	778	1652	100.0%

Table 5.4: Elements of the positive and negative set for each promoter group. The positive set always kept all elements, dropping instances from the negative set if necessary.

The inputs for the neural networks are the four vectors R (eq. 5.3). The number of elements in each vector is the number of eigenvectors for that group. Namely, 72 for the `tata-cpg-`, 69 for the `tata-cpg+`, 28 for `tata+cpg-` and 16 for `tata+cpg+`. As can be noticed, the number of eigenvectors increases consistently with the number of elements in the group. Fig. 5.2 contains a diagram with the data transformations that are performed in order to build the classifier.

Weka 3.4.13 was used for this task, run in commandline mode. The classifier was set to MultilayerPerceptron, and the other parameters were left with the default values except: autobuild True, decay False, nominalToBinaryFilter False, normalizeAttributes False, reset False. The training data was set to cross-validate in 10 folds.

An initial combination of parameters for each network was obtained by brute force. Combinations were computed by varying the learning rate between 0.1 and 1, step 0.1; the momentum between 0.1 and 1, step 0.1; and the number of hidden layers between 1 and 3. That methodology allows us to assert that our network quality is not far away from the best possible network for the given data. Networks with more than 3 hidden layers were discarded because of the computational cost and the marginal (if any) improvement in accuracy they presented.

As the quality measure we looked at the absolute number of correctly classified instances. The results for these networks were analyzed and the best three networks for each group were tweaked by hand, observing the impact of each variable in the resulting. Besides these variables, also the number of neurons in each layer and the training time were tested.

The four neural networks compose ProStar2's classifier, acting as the class predictor for promoter sequences. The models for each network can be found in the results section (6.2.4).

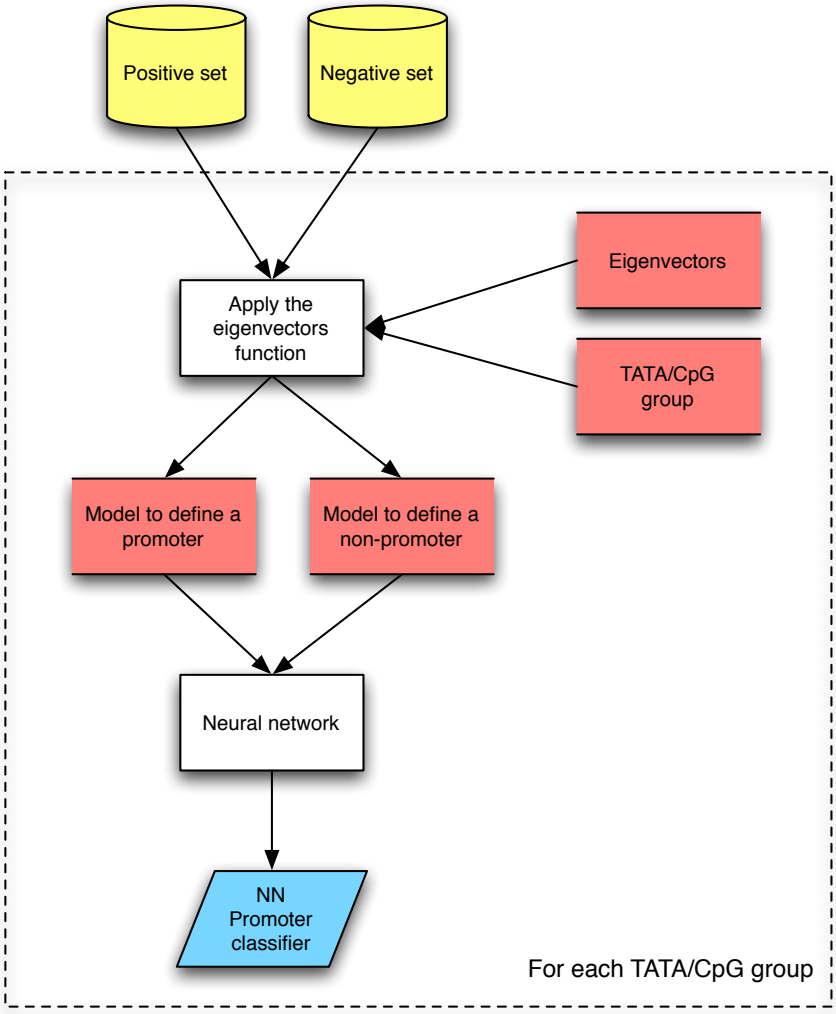


Figure 5.2: Data models for the predictor training.

5.5 ProStar 2.0

The new version of ProStar scans genomes in 2000-nts windows, moving forward 500-nts at each step. For every window we assume that the TSS would be placed in position +1000, the position where the TSS was located in the training set.

We first classify this target sequence as CpG+/CpG- and TATA+/TATA-. Once the promoter type is assigned to our target, we compute the appropriate physical descriptors with DNAlive scripts. Then the eigenvectors function is applied, and the resulting array is classified with the specific Neural Network. A graphical image of this process is depicted in fig. 5.3.

ProStar 2 only requires the user to input a DNA sequence, and it automatically performs all the calculations. Afterwards, it outputs the predicted class (promoter or non-promoter) and the confidence value for its prediction.

Our method is not available to predict the strand of the promoter, so the software will add an extra prediction with a negative strand to every promoter, doubling the amount of predictions.

5.6 Evaluation of the results

For evaluating the results, we looked at the results for the 10-fold cross-validation of each neural network. Weka outputs many accuracy descriptors, and we used the following:

- Correctly classified instances, which is the absolute accuracy, as it aggregates all the correct predictions.
- True positive, True negative, False positive and False negative percentages, as depicted in the confusion matrix.
- **Relative absolute error** . This measure is a ratio of the mean absolute error of the learning algorithm over the mean absolute error obtained by predicting the mean of the training data. In some cases, the percentage of correctly classified instances can be high because the predictions for the largest group are good, but smaller groups get worse predictions.

True Positives (TP) are promoters which were classified as such by the network. **False Positives** (FP) are non-promoters which were classified as promoters. **False Negatives** (FN) are promoters which were classified as non-promoters, and **True Negatives** (TN) were non-promoters correctly classified as such. The accuracy is defined in equation 5.4

$$\frac{TP + TN}{\#Instances} \tag{5.4}$$

As ProStar 1 does not cluster promoters by TATA/CpG, we will compare its global accuracy to each of our groups.

5.7 Concluding remarks

In this section we have introduced a methodology composed of many AI techniques. First, in order to increase the specificity of the predictions, the promoters were split in four groups. Then, to reduce input noise, we narrowed

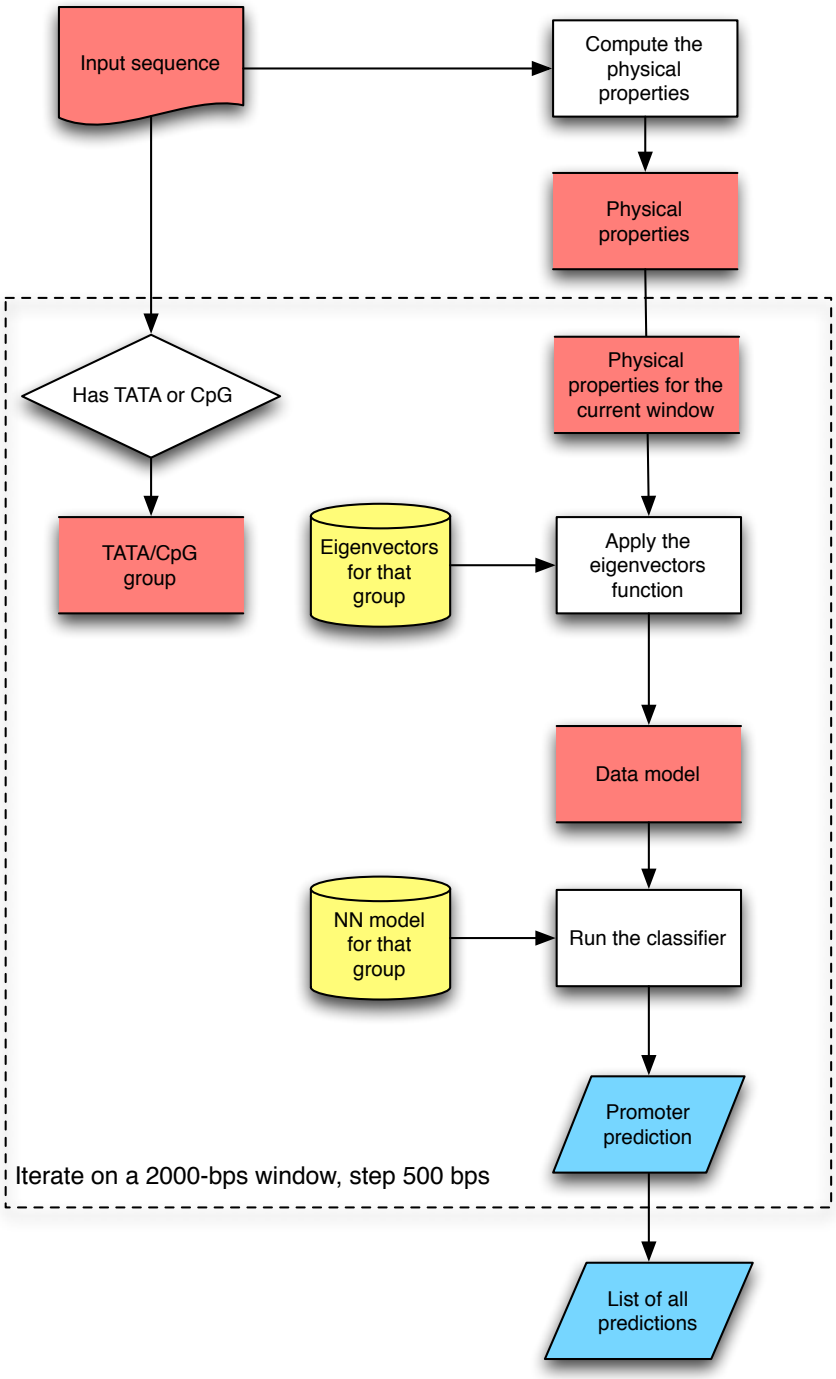


Figure 5.3: ProStar2 algorithm.

the number of descriptors with the aid of two statistical techniques. Finally, a neural network was built for each group. Each network was fed with the same number of instances for both classes, when possible.

As a result, the ProStar 2 software scans the input sequence in windows of the same size as the training set, stepping every 500 bases. It then classifies each window into the promoter or non-promoter class, and outputs each prediction with the confidence value.

To evaluate the results, we will compare the performance of each of the four networks to ProStar 1 (Results, table 6.4).

Chapter 6

Results

6.1 DNAlive

DNAlive has had a big impact in researchers, corroborated by the publication of a paper and two oral communications in international conferences[1]. The web page is accessed every day, especially for crossing ProStar's predictions with gene information at the UCSC Genome Browser.

DNAlive was developed to give a complete description of the physical properties of genomic DNA in a simple way, thus providing data that can be easily understood by non-structural experts. Among others, DNAlive allows the user to:

- Determine potential correlations between genome annotations (such as transcription start sites, exons, splicing sites, ...) and a battery of 29 physical descriptors of DNA,
- Find out the most stable 3D structure of long genome fragments using sequence-dependent average helical parameters, and, when available, experimental structural data on DNA-protein complexes,
- Perform a dynamic analysis of chromatin fiber exploring the range of deformability sampled during trajectory and the possibility of the formation of transient proteinprotein complexes, and
- Display structural parameters of DNA in the context of associated functional features obtained form several public databases.

Figure 6.1 depicts a screen shot of a DNAlive session in progress, and appendix A contains the published version of the manuscript.

6.2 ProStar 2

Many steps were involved in the new predictor methodology for ProStar 2. The next sections present the individual results for each of the phases.

[1] *DNAlive: A tool for a realistic representation of the DNA at a genomic scale.* VII Jornadas de Bioinformtica, Valencia, February 2007. *Representing the genomic DNA*. Grand challenges in computational biology, Barcelona, June 2008

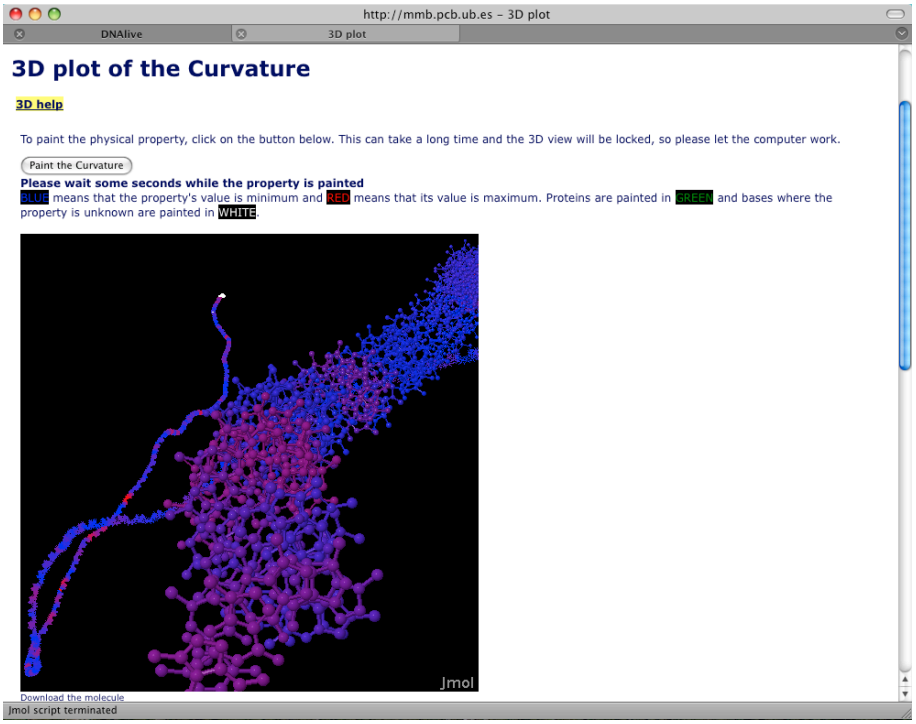


Figure 6.1: Screen shot of a DNAlive session. The user has selected a 3D view of a DNA sequence, and the Curvature descriptor is painted in a blue-to-red scale.

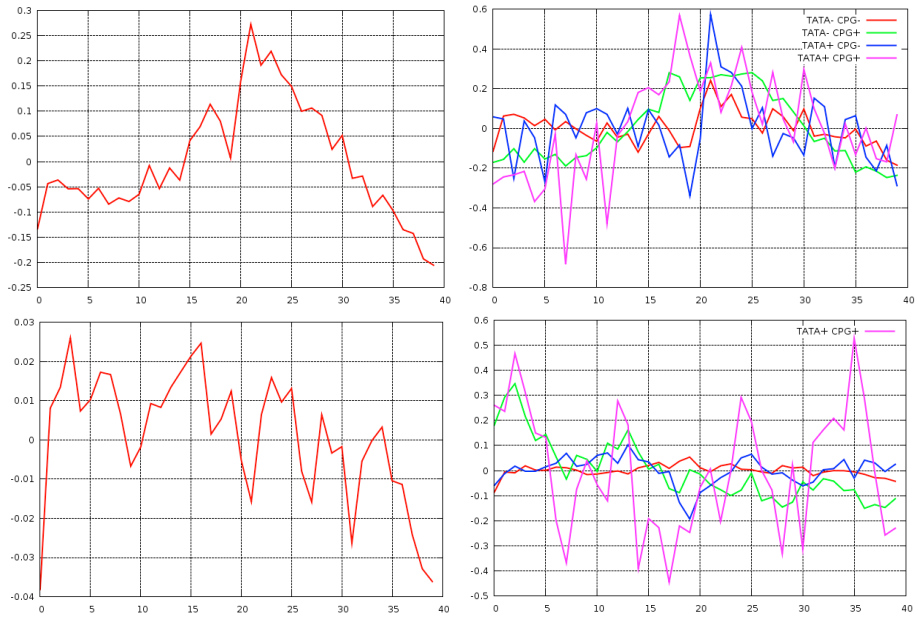


Figure 6.2: Plot for the CpG property on the positive (top) and negative (bottom) sets. The images on the left are the average for all promoter groups, while the images on the right have each group separately. A clear signal can be appreciated for the promoters, while the negative set only contains noise. The negative `tata+cpg+` (pink) group has higher variations because it contains less elements.

6.2.1 DNA descriptors signal analysis

Five plots were generated for each descriptor, one per promoter group and another for the aggregate of all the groups. Then, this was repeated for the negative set, only to check that there is only noise when the promoter is not present.

Figure 6.2 illustrates the results for the CpG property, which are very interesting. The aggregated signal shows a clear peak at the TSS position. Surprisingly, when looking at the split graphs, it is the non-cpg ones which contain this peak (the blue one and the red one).

A CpG island (lines `CPG+`) has a very specific definition and requirements (sec. 5.2.1). The results imply that, even if there are no CpG islands present, the CpG% content alone is a good indicator for the presence of a TSS (position 20 in the x-axis of the graph). Furthermore, one can infer that the presence of a CpG island in a 2000-bps sequence might reveal a promoter, even if the large CpG content is not exactly positioned near the TSS.

Like CpG, many physical properties have clear signals on the TSS. Another example is the curvature (fig. 6.3), which is displayed for illustration purposes. Almost every physical property has a clear peak at the TSS, but some properties work better for some promoter groups.

These results are very promising, as we did not expect to get such clear

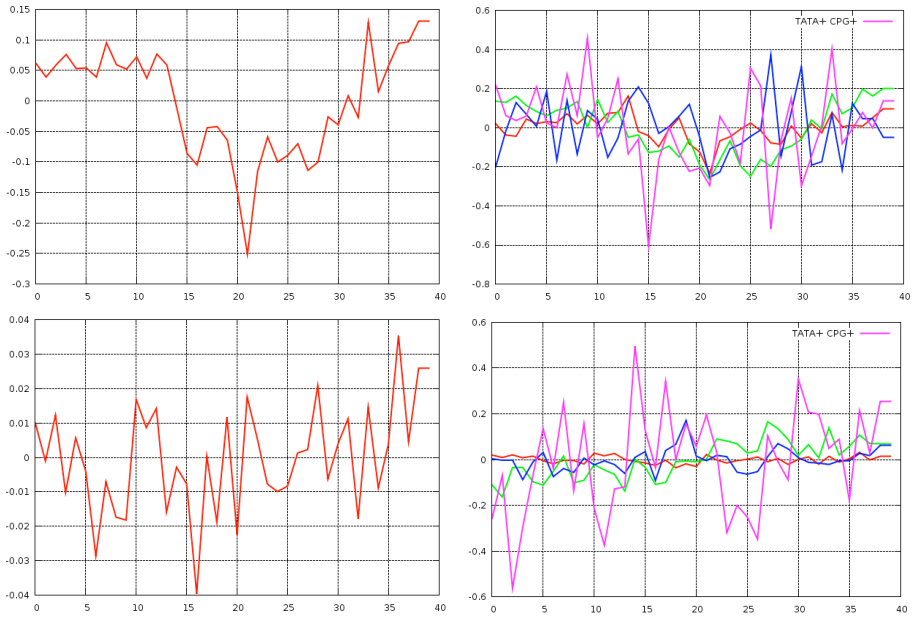


Figure 6.3: Plot for the Curvature property on the positive (top) and negative (bottom) sets. The images on the left are the average for all promoter groups, while the images on the right have each group separately. Again, a clear signal can be appreciated for the promoters, but not in the negative set.

TSS signals with that many properties. With this data, we can conclude that the DNA descriptors can carry promoter signals, and that this signal behaves differetly for each promoter group.

After looking at the graphs, we performed the numerical analysis of the signals.

6.2.2 Analysis of the physicochemical properties

The correlation analysis resulted in four tables, one per promoter group, displaying the Pearson correlation matrix between properties. In these tables, correlations higher than 90% appear in red, between 20% and 90% in yellow, and between 0% and 20% in white. For simplicity, only one of them is included in this document, arbitrarily selected, the one for the positive set `tata+cpg-` group, in the figure 6.4.

Also, the tables for the negative set are not shown. That is because the correlation values were very similar, with a variation range of 10% at most, thus concluding that physical properties are independent of the presence of a promoter. Furthermore, the tables for each promoter group are very similar, even though there are small differences.

For example, crossing CpG islands (descriptor `cpg`) with the six helical parameters (`krise ...ktwist`), shows that some of the parameters are more correlated with the groups which have CpG islands (`krise`), and some with the groups without them (`ktilt`).

Furthermore, at least half of the properties are highly correlated with the rest: `basestack`, `bendstiff`, `denaturation`, `dupldisrfreeen`, `duplstabfreeen`, `meltingtm`, `nuclpos`, `propwtist`, `protdeform`, `stability`, `stacking`, `z-dna`. This means that their values are probably a combination of some of the other properties, or just that the biological interpretation for all of them is very similar.

Figure 6.5 contains an image with some of the correlations, and besides the graph in the foreground, one can see many highly correlated plots in the background.

6.2.3 Principal components analysis

Once we built the descriptors matrix for all the input sequences, the PCA outputted a number of eigenvectors for each promoter group (tab. 6.1). Each of them was configured to keep 80% of the initial information. As expected, the larger the number of input elements, the higher the number of eigenvectors.

Group	Elements	Eigenvectors
`tata+cpg+`	30	17
`tata+cpg-`	60	29
`tata-cpg-`	377	73
`tata-cpg+`	407	70

Table 6.1: Resulting eigenvectors after the PCA

Table 6.2 contains part of the first three eigenvectors for the `tata-cpg+` group. In general, the first three eigenvectors carry about 20% of the original

Figure 6.4: Pearson correlation matrix for the TATA+ CpG- group.

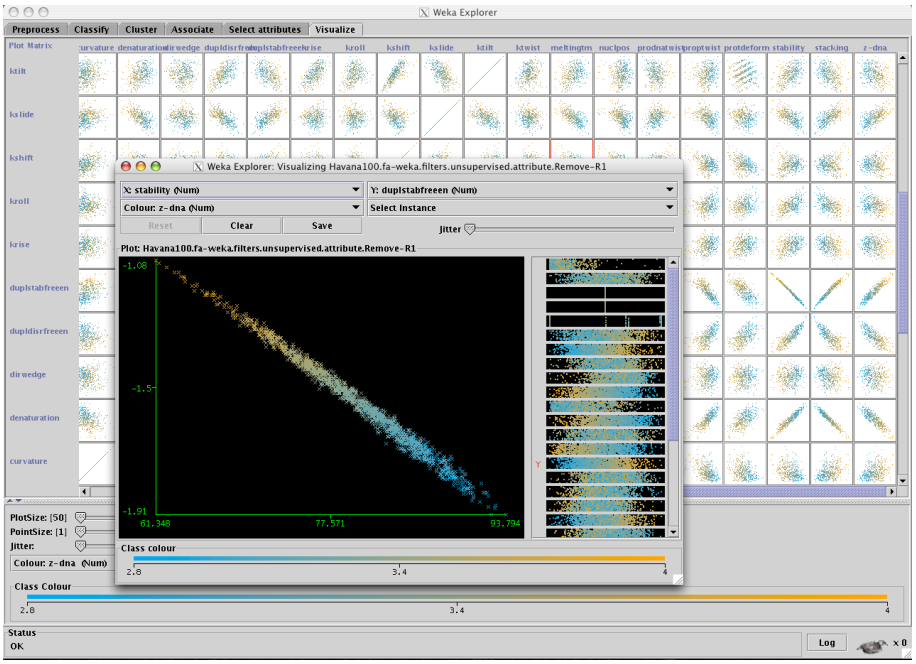


Figure 6.5: Weka plotting the correlation between Stability and Duplex Stability Free Energy.

information, and each one is a combination of weighted DNA properties/location (sec. 5.3.2). The first 10 items usually carry between 6% and 7% of the total weight, and they share the sequence position.

#	Weight	Elements
1	10.478%	$6.4\% * denaturation_{1000} - 6.4\% * stacking_{1000} +$ $+6.3\% * meltingtm_{1000} - 6.3\% * duplstabfreeen_{1000} + \ldots$
2	8.204%	$-6.4\% * denaturation_{1850} - 6.4\% * meltingtm_{1850} +$ $+6.4\% * stacking_{1850} - 6.4\% * stability_{1850} + \ldots$
3	4.026%	$-6.7\% * meltingtm_{800} - 6.7\% * stability_{800} +$ $+6.7\% * zdna_{800} + 6.6\% * stacking_{800}$

Table 6.2: The first three eigenvectors extracted from the `tata-cpg+`'s PCA

6.2.4 Neural network

Table 6.3 illustrates the variables that were used to build the neural network models. The methodology is detailed in section 5.4.

Results are evaluated as defined in sec. 5.6. Table 6.4 presents the accuracy results for each of ProStar2's promoter groups, versus ProStar1's global accuracy, as the latest does not cluster the promoters. ProStar1's accuracy (69%) has been improved only for the `tata+cpg+`, which predicts promoters with an

Group	Neurons / HL	Learning rate	Momentum	Training time
tata+cpg+	17, 17	0.5	0.2	500
tata-cpg+	70	0.2	0.1	500
tata-cpg-	73	0.2	0.4	500
tata+cpg-	29, 29, 29	0.3	0.4	500

Table 6.3: Neural network models for each promoter group. The groups are sorted by their accuracy (see tab. 6.4). The second column "Neurons per hidden layer" represents both the number of layers and the neurons in each one, separated by a comma. Note: in all cases, the number of neurons corresponds to the number of eigenvectors for that group.

accuracy of 79%. The other groups' accuracy is near 60% with very high values for the relative error.

The **relative absolute error** (column RA Error) is about 50% in the best case, and near 80% in the rest. This means that, in the worst case, the prediction power is only a bit better than a **ZeroR** prediction (i.e. the mode of all values).

Group	# Instances	TP	FP	FN	TN	Accuracy	RA Error
tata+cpg+	53	25	6	5	17	79.2453%	52.4784%
tata-cpg+	665	259	116	118	172	64.8120%	74.6566%
tata-cpg-	814	254	153	164	243	61.0565%	80.1061%
tata+cpg-	120	36	24	26	34	58.3333%	85.8145%
ProStar 1	1652	671	311	201	467	68.8861%	Unknown

Table 6.4: Accuracy results for ProStar 2 versus ProStar 1 . The first column indicates ProStar2's promoter group, and the aggregated for ProStar 1. ProStar1's results were gathered with a different methodology, which did not output the RA Error.

6.2.5 Final software

ProStar 2 is available as a web page at `http://mmb2.pcb.ub.es/proStar2`. It is quite simple and it did not require any engineering process like DNAlive. Instead, to emphasize the one-step prediction process, we designed a very minimalist interface. A screen shot of a ProStar 2 session is depicted at 6.6.

The interface is easy to use and no additional software is required other than a web browser. The web page launches ProStar 2 in the MMB group application server, and outputs the prediction when it is done.

The predictions of ProStar2 are not yet included in DNAlive, as we want to keep the currently published version of the page until it undergoes a major upgrade. No web service is available, for the same reason.

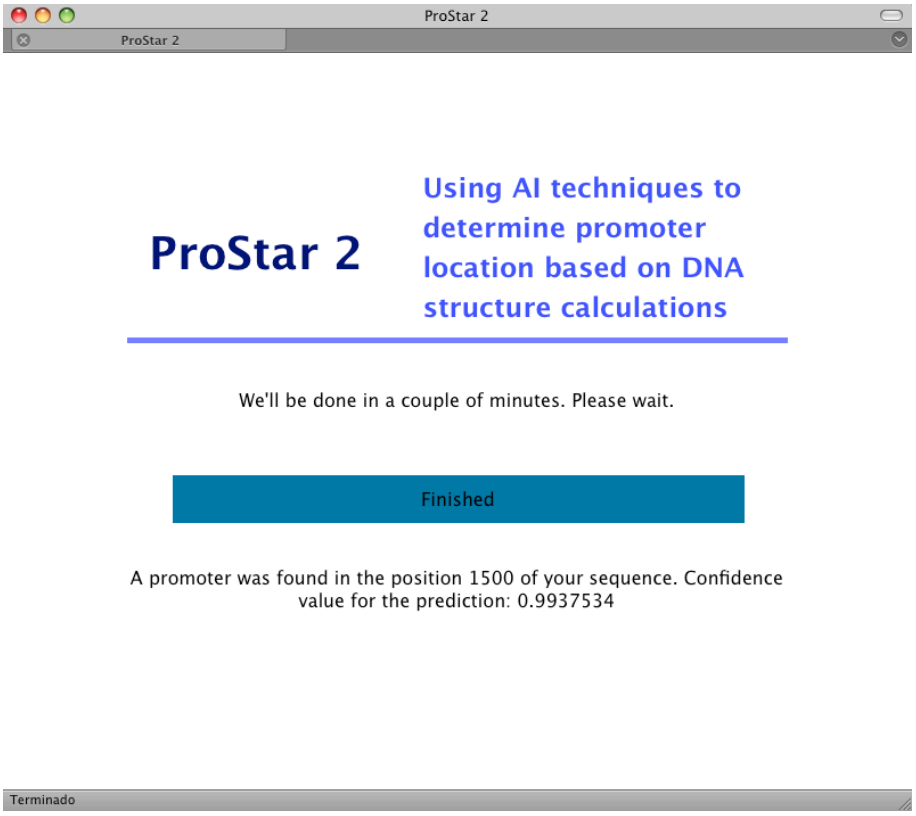


Figure 6.6: Screen shot of a ProStar 2 session. The user has input a DNA sequence and the software has predicted a promoter at the position 1500.

Chapter 7

Conclusions

In chapter 1.1 we presented the dual objectives for this thesis: creating a multi-interface platform to compute DNA descriptors, and implementing a new version for the ProStar promoter predictor, using AI techniques.

In the next two sections we will analyze the results for each project and compare them in detail to the expected objectives.

7.1 DNAlive

Huge databases are of no use without graphical tools to visualize the data and modular algorithms that can process jobs in batches.

Cross-referencing data is a very difficult work, sometimes even impossible, and at the current data growth rate researchers are now realizing that this might be the last opportunity to classify it, now that we still have the required computational power.

DNAlive was meant to be a simple interface for DNA description algorithms, but it has turned out to be a step forward into building tools whose purpose is to help users manage the available information. The implementation of the physical properties was derived from published work, data sources from the internet are used to cross-reference data, and if it is available, we use the annotated data from the UCSC Genome Browser to plot our tracks. 3D molecules and animations are displayed in a Jmol applet, and our web services call other web services to retrieve data and launch calculations.

To sum up, even if DNAlive has little Artificial Intelligence in it, it turns to be an interesting project for the degree in Computer Engineering, and surely will have more impact than ProStar 2.

7.2 ProStar 2

ProStar 2 has undergone a lot of changes from ProStar 1. Analyzing the methodology, it can be considered a different software rather than an evolution of the first one. The three-step process has forced us to make some decisions, which undoubtably have influenced the final results.

Some conclusions can be drawn from the obtained results:

- Promoter division in four groups. TATA boxes and CpG islands are the most popular features for defining promoters, and the resulting clusters' physical properties behaved different. This might mean that these features are good for promoter clustering, although maybe not the best ones.

- Sequence extraction. We arbitrarily decided to select the 2000 nucleotides surrounding each TSS. Looking at the signal for many properties, it is clear that selecting about 1000-nts sequences might have reduced noise, as the 500 initial and final nucleotides seem to carry no signal.

- Descriptors signal analysis. Most of the descriptors have a very clear peak at the TSS. However, not all promoter groups benefit from the same descriptors. In the Results chapter we described how the `tata+cpg-` group had a very distinctive signal for the CpG descriptor, but this signal was not that clear for the Curvature descriptor.

- Parameter correlation. In order to be computationally useful, 2000-nts sequences were reduced to 27 descriptors of 40 values. We do not believe that there is much information loss, as nearby nucleotides have similar physical behaviors. Despite the easily visible correlations, all the descriptors were used to train the neural network.

- Principal components analysis. We let the PCA decide the weight for each descriptor and position, hoping that it would be smart enough to select the most relevant ones. Looking at the resulting eigenvectors by hand, some positions far away from the TSS —that is, with no relevant signal— weighted more than what was expected.

- Predictor training. A MultilayerPerceptron was used as a classifier, and because it is very versatile, after analyzing the results, it seems that this was not a bad choice. Nonetheless, they tend to overfit and they are difficult to tune. Again, we performed a brute force initial approach, and the final results have been manually tuned based on the best networks. However, as might have been expected, the networks with less inputs didn't work better (see table 6.4).

Having descriptors which plot such clear signals, we should think why ProStar2's predictions are not that accurate. We wanted to automate the weight calculus for each descriptor with the PCA, so we did not filter them manually, exchanging automation for accuracy. Parameters should have been filtered by hand, keeping only the descriptors which had a clear signal (see sec. 6.2.1). Still, we believe that the methodology presented in this thesis can be fine tuned in order to increase other groups' accuracy.

For all that, we think that the area which can be more improved on is the Principal Components Analysis. The PCA cannot be blindly trusted to reduce noise. Instead of letting it extract 72 eigenvectors from 1,080 position/descriptors, it would be better to reduce the descriptors to about 100 and then get more specific eigenvectors. Another approach to reduce the number of eigenvectors is to keep less than 80% of the original information, say, about 50%.

The initial objective for ProStar2, improving ProStar1 predictive power, has been reached only for the small `tata+cpg+` promoter group (table 6.4). This

also suggests that clustering the promoters beforehand improves accuracy, but another clustering based on different properties could be more appropriate.

7.3 Future work

DNAlive can be updated when there are significant changes to the DNA descriptors. Also, it will adopt new technologies that allow researchers to visualize more data in a web browser, for example, a Flash-based molecular viewer instead of a Java one.

Furthermore, a new version is planned to integrate cross-species information from the Evolutionary Genomics[1] group at the IMIM.

Regarding ProStar 2, we have presented how promoter detection is a very complex problem that cannot be tackled with simple mathematical functions on energy values or text stemming techniques.

Our next steps will follow the ideas presented above. First of all, we will analyze the descriptors' correlations and signals per group, repeating the PCA with less data and creating new neural networks with less input parameters. Then, if the results do not improve, we will try to skip the PCA step and manually create a set of weighted inputs for the network.

We are reaching a point where there is too much data available, and researchers do not know where it can fit. Brute force techniques are computationally impossible, and blind searches do not work. Machine learning algorithms drop good results, but they need strong noise filters before they can be fed with data.

[1] `http://evolutionarygenomics.imim.es/`

Bibliography

[ABLC98] J F Allemand, D Bensimon, R Lavery, and V Croquette. Stretched and overwound DNA forms a pauling-like structure with exposed bases. *Proc Natl Acad Sci USA*, 95(24):14152–7, Nov 1998.

[AD07] P Akan and P Deloukas. DNA sequence and structural properties as predictors of human and mouse promoters. *Gene*, Jan 2007.

[ASB+08] T Abeel, Y Saeys, E Bonnet, P Rouze, and Y Van de Peer. Generic eukaryotic core promoter prediction using structural features of DNA. *Genome Res*, Jan 2008.

[ASRdP08] T Abeel, Y Saeys, P Rouze, and Y Van de Peer. ProSOM: Core promoter prediction based on unsupervised clustering of DNA physical profiles. *Bioinformatics*, page 9, May 2008.

[BBB+06] Vladimir B Bajic, Michael R Brent, Randall H Brown, Adam Frankish, Jennifer Harrow, Uwe Ohler, Victor V Solovyev, and Sin Lam Tan. Performance assessment of promoter predictions on ENCODE regions in the EGASP experiment. *Genome Biol*, 7 Suppl 1:S3.1–13, Jan 2006.

[BD98] R Blake and S Delcourt. Thermal stability of DNA. *Nucleic Acids Research*, 26(14):3323–3332, 1998.

[BFBM86] K Breslauer, R Frank, H Blöcker, and L Marky. Predicting dna duplex stability from the base sequence. *Proceedings of the National Academy of Sciences of the ...*, Jan 1986.

[BMHT91] A Bolshoy, P McNamara, R Harrington, and E Trifonov. Curved DNA without AA: experimental estimation of all 16 DNA wedge angles. *Proceedings of the National Academy of Sciences of the ...*, Jan 1991.

[BS03] Vladimir B Bajic and Seng Hong Seah. Dragon gene start finder: an advanced system for finding approximate locations of the start of gene transcriptional units. *Genome Res*, 13(8):1923–9, Aug 2003.

[BSC+02] Vladimir B Bajic, Seng Hong Seah, Allen Chong, Guanglan Zhang, Judice L Y Koh, and Vladimir Brusic. Dragon promoter finder: recognition of vertebrate RNA polymerase II promoters. *Bioinformatics*, 18(1):198–9, Jan 2002.

[BSC+03] Vladimir B Bajic, Seng Hong Seah, Allen Chong, S P T Krishnan, Judice L Y Koh, and Vladimir Brusic. Computer model for recognition of functional transcription start sites in RNA polymerase II promoters of vertebrates. *J Mol Graph Model*, 21(5):323–32, Mar 2003.

[BSSP95] I Brukner, R Sanchez, D Suck, and S Pongor. Trinucleotide models for DNA bending propensity: comparison of models based on DNaseI digestion and .... *J Biomol Struct Dyn*, Jan 1995.

[BTC+06] Vladimir B Bajic, Sin Lam Tan, Alan Christoffels, Christian Schönbach, Leonard Lipovich, Liang Yang, Oliver Hofmann, Adele Kruger, Winston Hide, Chikatoshi Kai, Jun Kawai, David A Hume, Piero Carninci, and Yoshihide Hayashizaki. Mice and men: their promoter properties. *PLoS Genet*, 2(4):e54, Apr 2006.

[Buc90] P Bucher. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. *J Mol Biol*, 212(4):563–78, Apr 1990.

[Cea05] P Carninci and et al. The transcriptional landscape of the mammalian genome. *Science*, 309(5740):1559–63, Sep 2005.

[CLH+96] P Cluzel, A Lebrun, C Heller, R Lavery, J L Viovy, D Chatenay, and F Caron. DNA: an extensible molecule. *Science*, 271(5250):792–4, Feb 1996.

[Cri70] F Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–3, Aug 1970.

[CW53] F Crick and J Watson. A structure for deoxyribose nucleic acid. *Nature*, Jan 1953.

[Dav03] Ramana V Davuluri. Application of FirstEF to find promoters and first exons in the human genome. *Current protocols in bioinformatics / editoral board, Andreas D Baxevanis [et al]*, Chapter 4:Unit4.7, May 2003.

[DGZ01] R V Davuluri, I Grosse, and M Q Zhang. Computational identification of promoters and first exons in the human genome. *Nat Genet*, 29(4):412–7, Dec 2001.

[DH02] Thomas A Down and Tim J P Hubbard. Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Res*, 12(3):458–61, Mar 2002.

[FSD+05] K Florquin, Y Saeys, S Degroeve, P Rouze, and Y Van de .... Large-scale structural analysis of the core promoter in mammalian and plant genomes. *Nucleic Acids Research*, Jan 2005.

[GB06] Samuel S Gross and Michael R Brent. Using multiple alignments to improve gene prediction. *J Comput Biol*, 13(2):379–93, Mar 2006.

[GGF87] M Gardiner-Garden and M Frommer. CpG islands in vertebrate genomes. *J Mol Biol*, Jan 1987.

[GPTO07] J Goñi, A Pérez, D Torrents, and M Orozco. Determining promoter location based on DNA structure first-principles calculations. *Genome Biology*, Dec 2007.

[GT81] O Gotoh and Y Tagashira. Stabilities of nearest-neighbor doublets in double-helical DNA determined by fitting calculated .... *Biopolymers*, Jan 1981.

[GZW95] A Gorin, V Zhurkin, and K Wilma. B-DNA twisting correlates with base-pair morphology. *J Mol Biol*, Jan 1995.

[HB05] Julian L Huppert and Shankar Balasubramanian. Prevalence of quadruplexes in the human genome. *Nucleic Acids Research*, 33(9):2908–16, Jan 2005.

[HC96] M El Hassan and C Calladine. Propeller-twisting of base-pairs and the conformational mobility of dinucleotide steps in DNA. *J Mol Biol*, Jan 1996.

[Hen06] C Henderson. Building scalable web sites. page 330, Jan 2006.

[HS89] P A Harbury and K Struhl. Functional distinctions between yeast TATA elements. *Mol Cell Biol*, 9(12):5298–304, Dec 1989.

[HZC90] P S Ho, G W Zhou, and L B Clark. Polarized electronic spectra of Z-DNA single crystals. *Biopolymers*, 30(1-2):151–63, Jan 1990.

[IM94] V I Ivanov and L E Minchenkova. The A-form of DNA: in search of the biological role. *Mol Biol (Mosk)*, 28(6):1258–71, Jan 1994.

[KB05] Aditi Kanhere and Manju Bansal. A novel method for prokaryotic promoter prediction based on DNA stability. *BMC Bioinformatics*, 6:1, Jan 2005.

[Ken02] W Kent.... BLAT—The BLAST-like alignment tool. *Genome Res*, Jan 2002.

[KKBB08] D Karolchik, R Kuhn, R Baertsch, and G Barber. The UCSC genome browser database: 2008 update. *Nucleic Acids Research*, Jan 2008.

[KSF+02] W James Kent, Charles W Sugnet, Terrence S Furey, Krishna M Roskin, Tom H Pringle, Alan M Zahler, and David Haussler. The human genome browser at UCSC. *Genome Res*, 12(6):996–1006, Jun 2002.

[KSW06] Edward Kawas, Martin Senger, and Mark D Wilkinson. BioMoby extensions to the taverna workflow management and enactment software. *BMC Bioinformatics*, 7:523, Jan 2006.

[LGK05] Haibo Liu, Jianmin Gao, and Eric T Kool. Helix-forming properties of size-expanded DNA, an alternative four-base genetic form. *J Am Chem Soc*, 127(5):1396–402, Feb 2005.

[LGLP92] F Larsen, G Gundersen, R Lopez, and H Prydz. CpG islands as gene markers in the human genome. *Genomics*, 13(4):1095–107, Aug 1992.

[LPKP01] V G Levitsky, O A Podkolodnaya, N A Kolchanov, and N L Podkolodny. Nucleosome formation potential of eukaryotic DNA: calculation and promoters analysis. *Bioinformatics*, 17(11):998–1010, Nov 2001.

[MVDC+08] Vincent Miele, Cedric Vaillant, Yves D'aubenton-Carafa, Claude Thermes, and Thierry Grange. DNA physical properties determine nucleosome occupancy from yeast to fly. *Nucleic Acids Research*, page gkn262v1, May 2008.

[OcLNR02] Uwe Ohler, Guo chun Liao, Heinrich Niemann, and Gerald M Rubin. Computational analysis of core promoters in the drosophila genome. *Genome Biology*, 3(12):RESEARCH0087, Jan 2002.

[OGL+98] W Olson, A Gorin, X Lu, L Hock, and V Zhurkin. DNA sequence-dependent deformability deduced from protein-DNA crystal complexes. *Proceedings of the National Academy of Sciences*, Jan 1998.

[ONGR01] U Ohler, H Niemann, Liao Gc, and G M Rubin. Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition. *Bioinformatics*, 17 Suppl 1:S199–206, Jan 2001.

[ORBM78] R ORNSTEIN, R REIN, D BREEN, and R MACELROY. An optimized potential function for the calculation of nucleic acid interaction energies. I- Base .... *Biopolymers*, Jan 1978.

[OSHN00] U Ohler, G Stemmer, S Harbeck, and H Niemann. Stochastic segment models of eukaryotic promoter regions. *Pacific Symposium on Biocomputing Pacific Symposium on Biocomputing*, pages 380–91, Jan 2000.

[PBCB98] A G Pedersen, P Baldi, Y Chauvin, and S Brunak. DNA structure in human RNA polymerase ii promoters. *J Mol Biol*, 281(4):663–73, Aug 1998.

[PMSS07] A Perez, I Marchan, D Svozil, and J Sponer. Refinement of the AMBER force field for nucleic acids: Improving the description of alpha/gamma .... *Biophysical Journal*, Jan 2007.

[RC96] R W Roberts and D M Crothers. Prediction of the stability of DNA triplexes. *Proc Natl Acad Sci USA*, 93(9):4320–5, Apr 1996.

[San98] J SantaLucia. A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc Natl Acad Sci USA*, 95(4):1460–5, Feb 1998. El paper de meltingtm.

[SCH96] J Sun, T Carestier, and C Hélène. Oligonucleotide directed triple helix formation. *Curr Opin Struct Biol*, Jan 1996.

[SCHH82] F Sanger, A Coulson, G Hong, and D Hill. Nucleotide sequence of bacteriophage lambda DNA. *J Mol Biol*, Jan 1982.

[SDT86] S Satchwell, H Drew, and A Travers. Sequence periodicities in chicken nucleosome core DNA. *J Mol Biol*, Jan 1986.

[SFL+95] M Sastry, R Fiala, R Lipman, M Tomasz, and D J Patel. Solution structure of the monoalkylated mitomycin C-DNA complex. *J Mol Biol*, 247(2):338–59, Mar 1995.

[SGLH97] J Sponer, H Gabb, J Leszczynski, and P Hobza. Base-base and deoxyribose-base stacking interactions in B-DNA and Z-DNA: a quantum-chemical study. *Biophysical Journal*, Jan 1997.

[SK95] A Sivolob and S Khrapunov. Translational positioning of nucleosomes on DNA: The role of sequence-dependent isotropic DNA .... *J Mol Biol*, Jan 1995.

[SK03] Stephen T Smale and James T Kadonaga. The RNA polymerase ii core promoter. *Annual Review of Biochemistry*, 72:449–79, Jan 2003.

[SKW00] M Scherf, A Klingenhoff, and T Werner. Highly specific localization of promoter regions in large genomic sequences by promoterinspector: a novel context analysis approach. *J Mol Biol*, 297(3):599–606, Mar 2000.

[SMHK01] Y Shen, K Musti, M Hiramoto, and H Kikuchi. Invariant asp-1122 and asp-1124 are essential residues for polymerization catalysis of family D DNA polymerase from pyrococcus horikoshii. *Journal of Biological Chemistry*, 276(29):27376–27383, Apr 2001.

[SNYH96] N Sugimoto, S Nakano, M Yoneyama, and K Honda. Improved thermodynamic parameters and helix initiation factor to predict stability of DNA duplexes. *Nucleic Acids Research*, 24(22):4501–4505, 1996.

[SS03] V V Solovyev and I A Shahmuradov. PromH: Promoters identification using orthologous genomic sequences. *Nucleic Acids Research*, 31(13):3540–5, Jul 2003.

[STS+01] Y Suzuki, T Tsunoda, J Sese, H Taira, J Mizushima-Sugano, H Hata, T Ota, T Isogai, T Tanaka, Y Nakamura, A Suyama, Y Sakaki, S Morishita, K Okubo, and S Sugano. Identification and characterization of the potential promoter regions of 1031 kinds of human genes. *Genome Res*, 11(5):677–84, May 2001.

[VAM+01] J Venter, M Adams, E Myers, P Li, and R Mural. The sequence of the human genome. *Science*, Jan 2001.

[vECLHP05] Titus S van Erp, Santiago Cuesta-Lopez, Johannes-Geert Hagmann, and Michel Peyrard. Can one predict DNA transcription start sites by studying bubbles? *Phys Rev Lett*, 95(21):218104, Nov 2005.

[VEH00] J M Vargason, B F Eichman, and P S Ho. The extended and eccentric E-DNA structure induced by cytosine methylation or bromination. *Nat Struct Biol*, 7(9):758–61, Sep 2000.

[WL02] Mark D Wilkinson and Matthew Links. BioMOBY: an open source biological web services proposal. *Brief Bioinformatics*, 3(4):331–41, Dec 2002.

Index

Appendix A

DNAlive publication

In this appendix appears the publication for DNAlive and three pages of the supplementary data where DNAlive is applied to find new genetic information.

BIOINFORMATICS APPLICATIONS NOTE Vol. 24 no. 15 2008, pages 1731–1732
doi:10.1093/bioinformatics/btn259

Structural bioinformatics

DNAlive: a tool for the physical analysis of DNA at the genomic scale

J. Ramon Goñi[1,2], Carlos Fenollosa[1,2,3], Alberto Pérez[1,2,3,4], David Torrents[1,2,5] and Modesto Orozco[1,2,3,4,*]

[1]Joint IRB-BSC Program on Computational Biology, Institute of Research in Biomedicine, Parc Científic de Barcelona, Josep Samitier 1-5, Barcelona 08028, [2]Barcelona Supercomputing Center, Jordi Girona 31, Barcelona 08034, [3]National Institute of Bioinformatics, Parc Científic de Barcelona, Josep Samitier 1-5, [4]Departament de Bioquímica, Facultat de Biología, Avgda Diagonal 647, Barcelona 08028 and [5]Institut Català per la Recerca i Estudis Avançats (ICREA), Barcelona, Spain

Received on March 27, 2008; revised on May 16, 2008; accepted on June 4, 2008

Advance Access publication June 9, 2008

Associate Editor: Alfonso Valencia

ABSTRACT

Summary: DNAlive is a tool for the analysis and graphical display of structural and physical characteristics of genomic DNA. The web server implements a wide repertoire of metrics to derive physical information from DNA sequences with a powerful interface to derive 3D information on large sequences of both naked and protein-bound DNAs. Furthermore, it implements a mesoscopic Metropolis code which allows the inexpensive study of the dynamic properties of chromatin fibers. In addition, our server also surveys other protein and genomic databases allowing the user to combine and explore the physical properties of selected DNA in the context of functional features annotated on those regions.

Availability: http://mmb.pcb.ub.es/DNAlive/ ; http://www.inab.org/

Contact: modesto@mmb.pcb.ub.es

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 INTRODUCTION

Massive genomic projects have revealed the sequence of nearly 50 eukaryotic genomes, including several mammals (among them, humans) and many more will become available in the coming years. So far, the annotation of these genomes has been nearly restricted to the identification and the one-dimensional location of functional features (mostly genes and their regulatory regions), without considering the structural parameters of their environment, which have been proven to be crucial for the functionality of DNA. Determining the structural properties of DNA and the combination of functional features is necessary to interpret and understand the functionality of genomes in a more complex, and therefore real, environment. The identification of these structural parameters allows scientists to consider different levels of accessibility of certain DNA regions to different proteins, such as transcription factors, polymerases and DNA methylases. For example, specific deformability or helical properties in a given region of DNA facilitate or impair the formation of nucleosomes hundreds of base pairs away, or can affect dimerization of two DNA-binding proteins which might be separated by thousands of bases in sequence. Different groups (Abeel *et al.*, 2008; Goñi *et al.*, 2007; Ohler *et al.*, 2001; Pedersen *et al.*, 2000; Singhal *et al.*, 2008) have demonstrated that regulatory regions in DNA display unusual physical properties, and in fact, two groups have recently proven independently (Abeel *et al.*, 2008; Goñi *et al.*, 2007) that eukaryotic promoters can be located with surprisingly good accuracy just analyzing simple physical descriptors of DNA, which confirms the existence of a hidden physical code that controls gene function. In summary, functional annotation needs to be complemented with physical data to understand the structure, dynamics and the general functionality of genomic DNA.

DNAlive has been developed to give a complete description of the physical properties of genomic DNA in a simple way, thus providing data that can be easily understood by non-structural experts. Among others, DNAlive allows the user to (i) determine potential correlations between genome annotations (such as transcription start sites, exons, splicing sites, …) and a battery of 29 physical descriptors of DNA (stability, helical descriptors, curvature, non-canonical B-DNA affinity, stiffness, …); (ii) find out the most stable 3D structure of long genome fragments (both naked DNA and DNA-protein complexes) using sequence-dependent average helical parameters, and, when available, experimental structural data on DNA-protein complexes; (iii) perform a dynamic analysis of chromatin fiber exploring the range of deformability sampled during trajectory and the possibility of the formation of transient protein–protein complexes and (iv) display structural parameters of DNA in the context of associated functional features obtained form several public databases. The tool is available as a web page and also as different webservices, which can be incorporated in user workflows (Supplementary Material).

2 IMPLEMENTATION

2.1 Entry data

The only mandatory input data for DNAlive is a DNA sequence in FASTA format or the genomic coordinates of a supported

*To whom correspondence should be addressed.

vertebrate genome. The program retrieves parameters from their internal databases (Supplementary Table 1) to determine physical profiles and to create a 3D structure of the naked DNA. Given a DNA sequence, the program determines potentially bound transcription factor binding sites (TFBS) by scanning the public TRANSFAC database (http://www.gene-regulation.com/) linked to PDB (http://www.rcsb.org/) and Uniprot databases (http://www.ebi.uniprot.org/). The selection of the complex of interest can be monitored externally by the user, who can force the generation of specific complexes (for example, nucleosomes, protein-multicomplexes, etc.).

2.2 Server workflow

Once a DNA sequence is entered (Fig. 1), the program computes the profile for the 29 physical properties available for the fiber (Supplementary Table 1). All properties are represented in a 2D plot using either the UCSC Genome Browser (http://genome.ucsc.edu) in combination with annotated genes whenever genomic coordinates for the genome are provided, or Gnuplot (Fig. 1 and Supplementary Fig. 1).

To combine the visualization of DNA physical properties with public annotations of the genome, coordinates of the input DNA sequence can be matched by running a search in our local Blat server (Kent, 2002). Although the user is able to annotate transcription factor PDB structures on specific positions of the DNA input sequence, we have implemented an automatic method to perform this step using the TFBS Perl library (Lenhard and Wasserman, 2002). The reconstruction of the average 3D structure of DNA is achieved using sequence-dependent base step parameters derived from accurate atomistic molecular dynamics (Pérez, 2007) and making use of a local adaptation of X3DNA (Lu and Olson, 2003) script (Fig. 1 and Supplementary Fig. 2). When structural information on protein–DNA complexes is available, modeled structures in the corresponding segment are substituted by the experimental geometries, and junctions are refined if required. The visualization of 3D structures is performed by integrating Jmol Java applets (http://www.jmol.org/) in the HTML page. All physical descriptors can be mapped into the 3D structure to favor the detection of potential correlations between conformation, functional annotations and physico-chemical properties (Fig. 1).

The server also includes unique tools for a rapid representation of chromatin dynamics, which, in extensive analysis performed in our laboratory on our database of more than 100 trajectories, showed a surprisingly high accuracy of the essential deformation pattern of DNA. The method uses a mesoscopic Metropolis Monte Carlo algorithm, where the geometry of each base pair is defined by three local rotations (roll, tilt and twist) and translations (slide, shift and rise), and the conformational energy is estimated from the deformation matrix using a harmonic model (Equation 1), where the index 'i' stands for one of the M base pair steps and the index 'j' stands for the six unique helical parameters (ξ) for each step. The equilibrium values for one helical parameter in a given base pair step type and (ξ_{ij}^0) and the associated deformation constant ($K_{i,j}$) were previously determined from molecular dynamics simulations (Pérez, 2007). Once a movement in helical coordinates is accepted by the Metropolis test, the corresponding Cartesian structure of the fiber is generated using an adaptation of X3DNA (Lu and Olson, 2003) for VIDEO visualization using JMOL Java applets in the HTML page (Supplementary Fig. 3). Basic manipulation and analysis of the trajectories and structure (rotations, translations, distance measurements,…) are allowed by the Jmol interface, which allows the determination of potential DNA-mediated protein-clusters.

$$E = \sum_{i=1}^{M} \sum_{j=1}^{6} K_{i,j} \left(\xi_{ij} - \xi_{ij}^0 \right)^2 \quad (1)$$

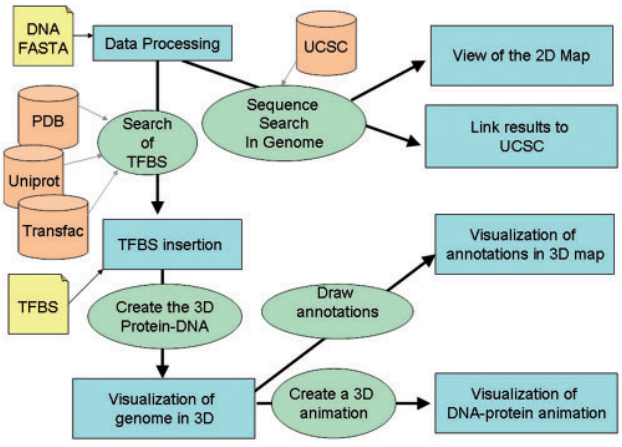


Fig. 1. DNAlive web server workflow diagram.

ACKNOWLEDGEMENTS

We thank the help of Agnes Noy, David Piedra, Henrique Proença and Joaquín Panadero as β-testers of the server.

Funding: This work has been supported by the Spanish Ministry of Education and Science (BIO2006-01602 and BIO2006-15036), the Spanish Ministry of Health (COMBIOMED network), the Fundación Marcelino Botín and the National Institute of Bioinformatics (Structural Bioinformatics Node).

Conflict of Interest: none declared.

REFERENCES

Abeel,T. *et al.* (2008) Generic eukaryotic core promoter prediction using structural features of DNA. *Genome Res.*, **18**, 310–323.

Goñi,J.R. *et al.* (2007) Determining promoter location based on DNA structure first-principles calculations. *Genome Biol.*, **8**, R263.

Kent,W.J. (2002) BLAT- the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.

Lenhard,B. and Wasserman,W.W. (2002) TFBS: computational framework for transcription factor binding site analysis. *Bioinformatics*, **18**, 1135–1136.

Lu,X.J. and Olson,W.K. (2003) 3DNA: a software package for the analysis, rebuilding and visualization of three-dimensional nucleic acid structures. *Nucleic Acids Res.*, **31**, 5108–5121.

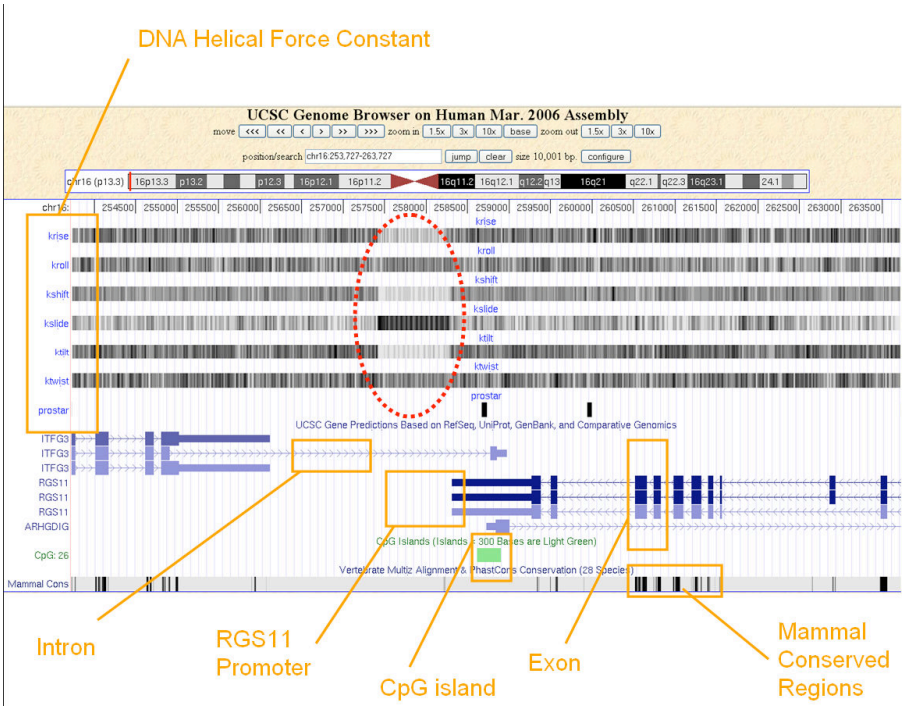
Ohler,U. *et al.* (2001) Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition. *Bioinformatics*, **17** (Suppl. 1), S199–S206.

Pedersen,A.G. *et al.* (2000) A DNA structural atlas for *Escherichia coli*. *J. Mol. Biol.*, **299**, 907–930.

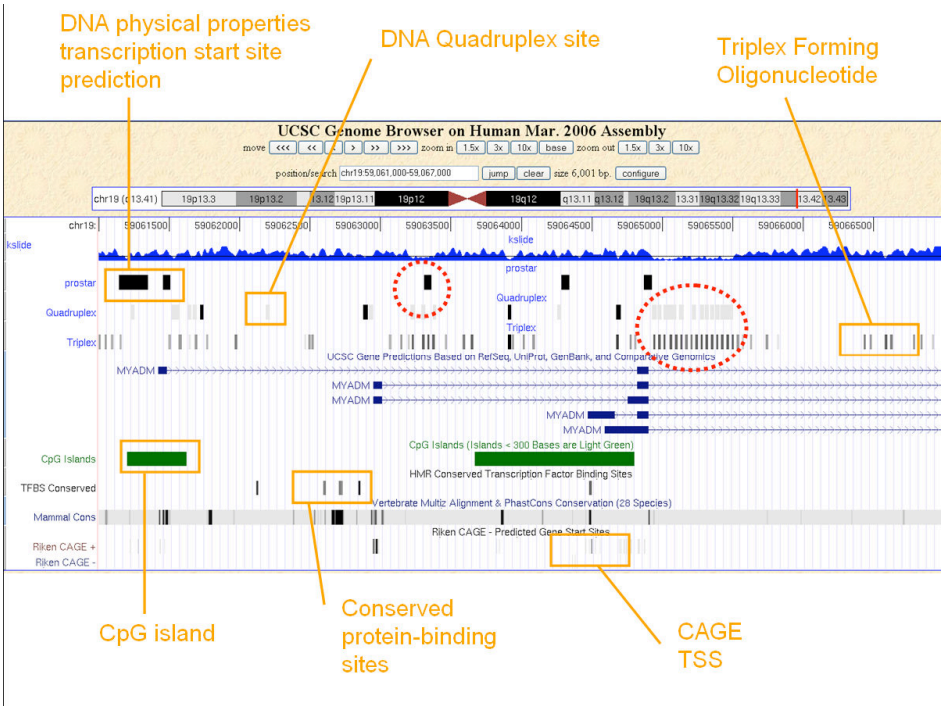
Pérez,A. *et al.* (2007) Refinement of the AMBER force field for nucleic acids. Improving the description of α/γ conformers. *Biophys. J.*, **92**, 3817–3829.

Singhal,P. *et al.* (2008) Prokaryotic gene finding based on physicochemical characteristics of codons calculated from molecular dynamics simulations. *Biophys. J.* [EPub ahead of print; DOI:10.1529/biophysj.107.116392].

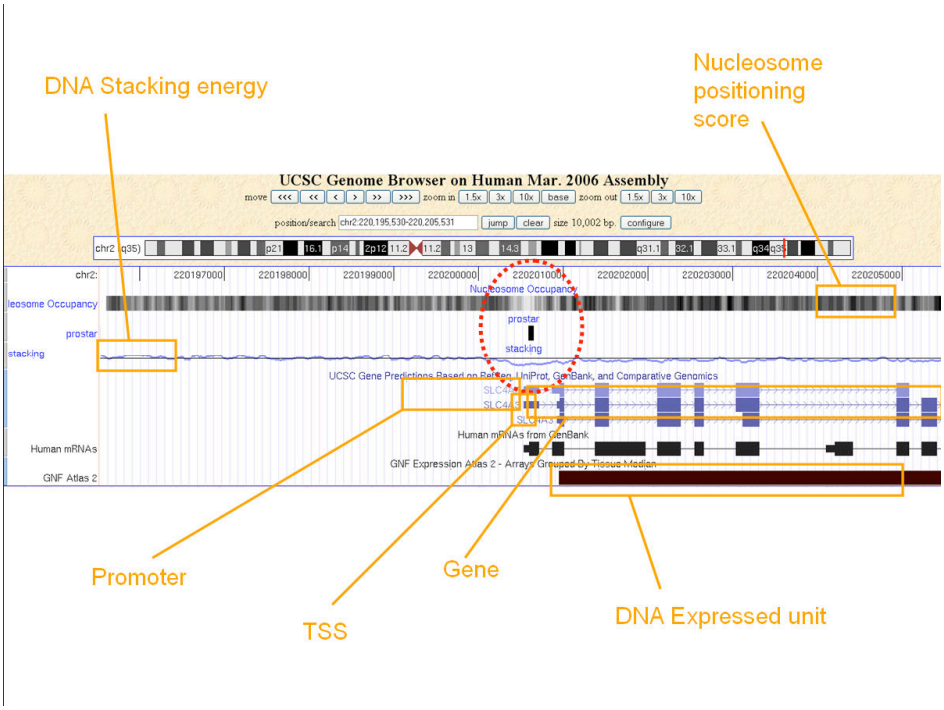
Example of physical properties in regulatory regions



Supplementary Figure 4. In the human telomeric region 16p13.3 (Chr16:253:727-263:727) the promoter of RGS11 shows a strong perturbation of DNA helical force-constants. Interestingly, the 5'UTR region overlaps with a 5'UTR region of a gene in the reverse strand (ITFG3). This increase of flexibility on Tilt and Shift (and decrease for Slide) is not associated to a CpG island.



Supplementary Figure 5. The gene MYADM has annotated multiple (but close) transcription start sites(TSS), revealing a complex regulatory mechanisms. The DNA-structural based algorith ProStar reveals the possible presence of a new TSS around position chr19:59,063,300. Although two CpG rules this region, the presence of Triple-helices and Qudruplex is very rich and may play a key role on regulation (see Goñi et al. 2006). This is more obvious around position chr19:59,065,000



Supplementary Figure 6. The TSS of SLC4A3 (a CpG promoter) is a nucleosome-free region. The sequences upstream of the 5'-end of genes normally show a low affinity for nucleosomes. DNAlive allow for the first time correlate structural parameters like DNA stacking energies with DNA high-order structures. A decrease of the stacking energy rise up the stability of the DNA double helix molecule.

Appendix B

Description of the physical properties

The following tables contain the description for each of the 29 physical properties and their references.

B.1 Unusual DNA conformation

Name	Description	Additional information	Reference
Z-DNA	Free energy values of Z-DNA transition	Derived from polarized electronic absorption spectra of single crystals of Z-form duplexes	[HZC90]
A-DNA	Probability derived from energy cost of the B-DNA to A-DNA cost	Physochemical parameters derived using the neighbor approach.	[IM94]
Triplex stability	Stability estimation of DNA triple helices	Parameters (enthalpies, Tm and free energies) obtained assuming neighbour approach in parallel triplexes. Experimental values derived from calorimetric studies of different triplex sequence at different pHs.	[RC96]
Quadruplex	Potential quadruplex sequence	Empirical method for prediction of G-DNA tetrads based on the strand stoichiometry, number of stacked tetrads, mutations/deletion and length and composition of loops.	[HB05]

Table B.1: Unusual DNA conformation

B.2 DNA disruption energy

Name	Description	Additional information	Reference
Base-pair stacking	Base-stacking energy	Derived from simple force-field energies using equilibrium geometries	[ORBM78]
Duplex disruption free energy	DNA disrupt energy	Parameters (enthalpies and entropies) obtained assuming neighbour approach. Experimental values derived from calorimetric studies of 19 DNA oligomers and 9 DNA polymers.	[BFBM86]
Duplex stability free energy	Thermodynamic free energy	Corrected-near neighbour parameters fit to experimental data (50) and cross-validated with other 15 oligos. Values fitted to melting curves.	[SNYH96]
Stacking energy	Stacking energy from quantum-chemical calculations	Derived from high quality quantum mechanical calculations (gas phase) on equilibrium geometries of the 10 unique DNA dimer steps.	[SGLH97]

Table B.2: DNA disruption energy

B.3 DNA 3DNA structure

Name	Description	Additional information	Reference
B-DNA twist	Twist angle torsion based on B-DNA crystals	Geometrical parameters derived from analysis of crystal databases (mostly slide and propeller twist analysis). They are obtained assuming neighbor approach.	[GZW95]
Curvature	Curvature based on twist, wedge, direction calculations from gel retardation experiments	Parameters derived from crystal structures of B-DNA and valid within the neighbour limit.	[BMHT91]
Direction Wedge	Direction of the deflection of the axial path of DNA	Parameters derived from crystal structures of B-DNA and valid within the neighbour limit.	[BMHT91]
Protein DNA twist	Twist angle torsion based on Protein-DNA complexes	Geometrical parameters derived from analysis of crystal databases of DNA-protein complexes within the neighbour approach.	[OGL$^+$98]
DNA propeller-twist	Propeller-twist base pair measure from crystallographic data	Geometrical parameters derived from analysis of crystal databases. Mobility is represented from deviations in the propeller twist/slide profile. Values are obtained assuming neighbour approach.	[HC96]

Table B.3: DNA 3DNA structure

B.4 DNA flexibility

Name	Description	Additional information	Reference
Bendability	Deformability based on DnaseI cutting frequencies	Parameters derved from DNase I digestion and nucleosome binding data and applied at the trinucleotide level.	[BSSP95]
Bending stiffness	Rigidity of the DNA helix	Method to predict nucleosomal translational position in terms of bending free energy computed from the near neighbour model.	[SK95]
Nucleosome preference	Nucleosome trinucleotide preference	Parameters derived to predict rotational preference of nucleosomes based on fitting to 177 sequences of chicken erythrocite core particles.	[SDT86]
Protein deformation	Deformability based on DNA-protein crystal structures	Deformation parameters derived from analysis of crystal databases of DNA-protein complexes within the neighbour approach.	[OGL$^+$98]

Table B.4: DNA Flexibility

B.5 DNA stability

Name	Description	Additional information	Reference
Denaturation	Denaturation equilibrium Parameters (enthalpies and entropies) obtained assuming neighbour approach.	Experimental values derived from high-resolution melting curves of synthetic DNAs inserted in pN/MCS plasmids.	[BD98]
Melting temperature	Melting temperature	Derived by fitting to melting profiles of restriction fragment of PHIx174 and fd phage DNAs. Near neighbor model is used.	[GT81]
Stability	DNA melting temperature derived from single set of parameters	Unified view of DNA melting. Near neighbor parameters obtained by fitting accurate calorimetric data.	[San98]

Table B.5: DNA stability

B.6 DNA non-linear dynamics

Name	Description	Additional information	Reference
Bubbles	Non-linear predictor of bubble formation on DNA	Derived from Peyrard-Bishop-Dauxois dynamics simulations using an ultrasimplified quasi-harmonic potential fitted to reproduce denaturalization curves of short hereogeneous DNA segments.	[vECLHP05]

Table B.6: DNA non-linear dynamics

B.7 PARMBSC0 Helical force constants

Name	Description	Additional information	Reference
Rise	Z-axis translational deformability	Derived from atomistic MD simulations in water for a small set of duplexes cointaining all unique dinucleotide steps. Helical force-constants are derived by inversion of the covariance matrix in helical space and assuming harmonic oscillations. The neighbor approach is used.	[PMSS07]
Roll	Y-axis rotational deformability		
Shift	X-axis translational deformability		
Slide	Y-axis translational deformability		
Tilt	X-axis rotational deformability		
Twist	Z-axis rotational deformability		

Table B.7: PARMBSC0 Helical force constants.

B.8 Regulation parameters

Name	Description	Additional information	Reference
ProStar	Prediction of promoter regions using DNA-physical properties	Ab initio promoter prediction of mammal genomes. The method used helical force constants as descriptors and is trained using known annotated promoters in humans.	[GPTO07]
Nucleosome potential	Nucleosome preferential occupancy estimator	Derived by a Monte Carlo code using nucleosome formation potentials obtained from trained dinucleotide parameters. Method was fit using a dataset of 141 nucleotide sequences which were experimentally annotated.	[LPKP01]

Table B.8: Regulation parameters

This document was printed on September 3, 2008